



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA AUTOMOTRIZ**

DESARROLLO DE UN SIMULADOR DE CONDUCCIÓN A TRAVÉS DEL PROGRAMA VISUAL STUDIO PARA LA EVALUACIÓN DE REACCIONES Y REFLEJOS EN ESTUDIANTES DE INGENIERÍA AUTOMOTRIZ DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, QUITO.

Trabajo de titulación previo a la obtención del
Título de Ingeniero Automotriz.

AUTORES : CHRISTIAN PAUL SANGUCHO RONDAL
STEVEN FERNANDO MORILLO OBANDO

TUTOR: DIEGO ANDRÉS DUQUE SARMIENTO

Quito - Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Christian Paul Sangucho Rondal con documento de identificación N° 1727009290 y Steven Fernando Morillo Obando con documento de identificación N° 1751159136 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 1 de agosto del año 2024

Atentamente,



Christian Paul Sangucho Rondal
1727009290



Steven Fernando Morillo Obando
1751159136

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Christian Paul Sangucho Rondal con documento de identificación No. 1727009290 y Steven Fernando Morillo Obando con documento de identificación No. 1751159136, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Desarrollo de un simulador de conducción a través del programa visual studio para la evaluación de reacciones y reflejos en estudiantes de Ingeniería Automotriz de la Universidad Politécnica Salesiana, Quito. ”, el cual ha sido desarrollado para optar por el título de Ingenieros Automotrices, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 1 de agosto del año 2024

Atentamente,



Christian Paul Sangucho Rondal
1727009290



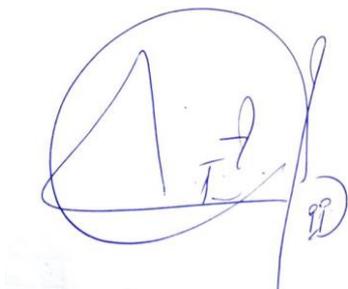
Steven Fernando Morillo Obando
1751159136

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Diego Andrés Duque Sarmiento con documento de identificación N° 1900611003, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN SIMULADOR DE CONDUCCIÓN A TRAVÉS DEL PROGRAMA VISUAL STUDIO PARA LA EVALUACIÓN DE REACCIONES Y REFLEJOS EN ESTUDIANTES DE INGENIERÍA AUTOMOTRIZ DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, QUITO. Realizado por Christian Paul Sangucho Rondal con documento de identificación No. 1727009290 y Steven Fernando Morillo Obando con documento de identificación No. 1751159136, obteniendo como resultado final el trabajo de titulación bajo la opción: Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 1 de agosto del año 2024.

Atentamente,



Ing. Diego Andrés Duque Sarmiento MSc.

1900611003

DEDICATORIA

Hoy, al finalizar este arduo camino de investigación y aprendizaje, deseo dedicar este logro a todas las personas que han sido fundamentales en mi trayectoria académica.

A mis padres, por su amor incondicional, su apoyo constante y por creer en mí incluso cuando yo dudaba de mis propias habilidades. Su sacrificio y dedicación me han brindado la oportunidad de llegar hasta este momento y nunca podré agradecerles lo suficiente.

A mi hermana, quien han sido mi inspiración y motivación. Sus logros personales y académicos me han impulsado a dar lo mejor de mí en cada etapa de este camino, y su cariño ha sido un soporte invaluable en los momentos de dificultad.

A mis amigos, por su compañía y por ser mi refugio en las jornadas agotadoras. Sus palabras de aliento, sus consejos y su presencia constante han sido la fuerza que me ha impulsado a seguir adelante cuando las cargas parecían abrumadoras.

A mis profesores y mentores, quienes han compartido su sabiduría y experiencia conmigo. Gracias por desafiarme a pensar de manera crítica, por alentarme a explorar nuevos horizontes y por guiarme en el proceso de convertirme en un pensador independiente y creativo.

Agradezco también a todos los participantes de mi estudio, cuya colaboración y contribuciones han sido esenciales para el desarrollo de mi investigación. Sin su participación y generosidad, este trabajo no habría sido posible.

Finalmente, dedico este logro a mí mismo. A lo largo de este viaje, he enfrentado obstáculos y desafíos que me han ayudado a crecer y a fortalecerme como persona. Mi determinación, perseverancia y pasión por el conocimiento me han llevado hasta aquí, y me siento orgulloso de todo lo que he logrado.

A todos ustedes, mi más profundo agradecimiento. Este logro no solo es mío, sino de todos aquellos que han dejado una huella en mi vida. Espero que este trabajo de investigación pueda contribuir de alguna manera al avance de la ciencia y a la mejora de nuestra sociedad.

Con gratitud y afecto,

Christian Sangucho

AGRADECIMIENTO

Aprovecho este espacio para expresar mi sincero agradecimiento a las personas que han sido parte fundamental de mi camino en esta etapa de mi vida.

En primer lugar, quiero agradecer a mi director de tesis, Diego Duque, por su guía experta, su paciencia y su constante apoyo a lo largo de este proceso. Sus conocimientos y experiencia han sido invaluable para el desarrollo de mi investigación y su dedicación me ha inspirado a dar lo mejor de mí en cada paso.

Quiero expresar mi gratitud a mis compañeros de laboratorio y a mis compañeros de estudios, quienes han sido una fuente constante de inspiración y colaboración. Sus discusiones y aportes han enriquecido mi proceso de aprendizaje y me han impulsado a ir más allá de mis límites.

Agradezco a la Universidad Politécnica Salesiana en la que he llevado a cabo mi investigación, así como a todos los profesores y personal administrativo que han brindado su apoyo a lo largo de mi carrera. Su compromiso con la educación y su labor incansable han sido fundamentales para mi formación académica.

No puedo dejar de mencionar a mis seres queridos y amigos cercanos, quienes han sido mi red de contención emocional y mi mayor apoyo durante estos años. Sus palabras de aliento, su paciencia y su comprensión han sido fundamentales para superar los desafíos y mantenerme motivado en los momentos más difíciles.

Por último, pero no menos importante, quiero agradecer a todas las fuentes de conocimiento que han nutrido mi investigación. A los autores, investigadores y académicos cuyos trabajos y publicaciones han sido referentes en mi campo de estudio, les estoy agradecido por su contribución al avance del conocimiento.

Sin cada una de estas personas, este logro no habría sido posible. Su apoyo, orientación y cariño han sido un regalo invaluable en mi vida.

De nuevo, mi más profundo agradecimiento a todos ustedes.

Steven Morillo

ÍNDICE GENERAL

ÍNDICE GENERAL.....	6
ÍNDICE DE TABLAS	10
RESUMEN.....	1
INTRODUCCIÓN	3
PROBLEMA	5
Objetivo General.....	9
Objetivos Específicos.	10
Marco Teórico.....	10
2.1 Evaluación de Habilidades en la Conducción.....	11
2.1.1 Habilidades cognitivas.....	11
2.1.2 Habilidades motoras	12
2.1.3 Métodos y herramientas convencionales utilizados para evaluar habilidades de conducción.....	12
2.2 Tecnología y Simuladores en la Conducción	14
2.3 Factores de Riesgo y Seguridad Vial	15
2.3.1 Seguridad vial	15
2.3.2 Factores de riesgo	16
2.4 Estudios Comparativos y Edad en la Conducción.	17
2.5 Normativas y Estándares en Conducción	17
2.6 Relevancia del Simulador Propuesto	19
CAPÍTULO 1	20
DESARROLLO DEL PROYECTO.....	20
1.1 Software	20
1.1.1. Visual Studio 2022	20
Depuración.....	23
1.1.2. Elaboración De Pantalla De Inicio con C#	24

1.2. Hardware.....	29
1.2.1. Volante y pedales.....	30
1.2.2. TwinWheel F1	30
1.2.3. Desarmado del volante	30
1.2.4. Pedales: Mecanismo	32
1.2.5. Interfaz: Conexión	34
1.3. Test de prueba de reacción simple.....	34
1.4. Programación del simulador de conducción	37
Control Form Pedalera C#.....	57
1.5. Test de resistencia a la monotonía.....	69
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS OBTENIDOS	80
2.1. Análisis e interpretación de resultados del test de reacción simple	81
2.2. Análisis e interpretación de resultados del test de resistencia a la monotonía	85
CONCLUSIONES	94
RECOMENDACIONES	95
REFERENCIAS BIBLIOGRÁFICAS.....	96
Presupuesto.....	99

ÍNDICE DE FIGURAS

Figura 1.1: Descargas de Visual Studio	21
Figura 1.2: Bienvenidos al programa	22
Figura 1.3: Visual Studio Ejemplo Autocompletado.	23
Figura 1.4: Visual Studio Corrección De Errores	24
Figura 1.5: Aplicación Windows Forms.	25
Figura 1.6: Cuadro Herramientas.	26
Figura 1.7: Boceto pantalla de inicio.	27
Figura 1.8: Identificación del “Button”.....	28
Figura 1.9: Creación del “Button”.....	28
Figura 1.10: Pantalla del test de reacción simple.....	29
Figura 1.11: Presentación de la pantalla principal con los botones de ingreso a los test... 29	
Figura 1.12: Volante y Pedal Simulador	30
Figura 1.13: Volante desarmado	31
Figura 1.14: Partes del volante a ser utilizado	32
Figura 1.15: Pulsadores del Simulador	32
Figura 1.16: Pedales Simulador.	33
Figura 1.17: Cable de audio que va conectado desde los pedales a la placa.....	34
Figura 1.18: Cable USB que se conecta al computador.....	34
Figura 1.19: Conexión de los cables	34
Figura 1.20: Pedales y maniobras con el pie derecho	36
Figura 1.21: Resultados del test (promedio)	37
Figura 1.22: Programación C#	38
Figura 1.23: Programación.....	49
Figura 1.24: Programación pedalera central	52
Figura 1.30: Programación.....	57
Figura 1.31: Programación public class	58
Figura 1.35: Programación.....	68
Figura 1.37: Declaración de las directivas.	68
Figura 1.43: Cierre del programa.	69
Figura 1.44: Botonera para el Test.....	70

Figura 1.45: Pantalla del Test de resistencia a la monotonía	71
Figura 1.46: Control del test a la resistencia a la monotonía	71
Figura 1.47: Botonera del simulador de conducción	72
Figura 1.48: Impresión 3B botonera parte superior	72
Figura 1.49: Impresión 3D botonera parte inferior	73
Figura 1.50: Mecanismo de la botonera.	73
Figura 1.51: Adaptación de la placa a la botonera	74
Figura 1.52: Verificación del pedal (antes).....	75
Figura 1.53: Verificación de la pedatera (después).....	75
Figura 1.54: Pulsador adaptado al pedal	76
Figura 1.55: Presentación del test de reacción simple	76
Figura 1.56: Presentación del test de Resistencia a la monotonía.....	77
Figura 1.57: Aplicación del Test de reacción simple	78
Figura 1.58: Finalización del test de reacción simple	79
Figura 1.59: Presentación final.....	79
Figura 2.1: Totalidad de participantes.....	85
Figura 2.2: Resultados de todos los participantes	93
Figura 2.3: Presupuesto	99

ÍNDICE DE TABLAS

Tabla 2.1: Resultado del participante 1 test de reacción simple.....	81
Tabla 2.2: Resultado del participante 2 test de reacción simple.....	81
Tabla 2.3: Resultado del participante 3 test de reacción simple.....	81
Tabla 2.4: Resultado del participante 4 test de reacción simple.....	82
Tabla 2.5: Resultado del participante 5 test de reacción simple.....	82
Tabla 2.6: Resultado del participante 6 test de reacción simple.....	83
Tabla 2.7: Resultado del participante 7 test de reacción simple.....	83
Tabla 2.8: Resultado del participante 8 test de reacción simple.....	84
Tabla 2.9: Resultado del participante 9 test de reacción simple.....	84
Tabla 2.10: Resultado del participante 10 test de reacción simple.....	85
Tabla 2.11: Resultado del participante 11 test de reacción simple.....	86
Tabla 2.12: Resultado del participante 12 test de reacción simple.....	86
Tabla 2.13: Resultados del test de reaccion simple	86
Tabla 2.14: Resultados del participante 1 test de resistencia a la monotonía.....	85
Tabla 2.15: Resultados del participante 2 test de resistencia a la monotonía.....	86
Tabla 2.16: Resultados del participante 3 test de resistencia a la monotonía.....	86
Tabla 2.17: Resultados del participante 4 test de resistencia a la monotonía.....	87
Tabla 2.18: Resultados del participante 5 test de resistencia a la monotonía.....	88
Tabla 2.19: Resultados del participante 6 test de resistencia a la monotonía.....	88
Tabla 2.20: Resultados del participante 7 test de resistencia a la monotonía.....	88
Tabla 2.21: Resultados del participante 8 test de resistencia a la monotonía.....	89
Tabla 2.22: Resultados del participante 9 test de resistencia a la monotonía.....	90
Tabla 2.23: Resultados del participante 10 test de resistencia a la monotonía.....	90
Tabla 2.24: Resultados del participante 11 test de resistencia a la monotonía.....	91
Tabla 2.25: Resultados del participante 12 test de resistencia a la monotonía.....	91
Tabla 2.26: Resultados de todos los participantes	92

RESUMEN

El presente proyecto investigativo posee como objetivo construir un Simulador de conducción de pruebas de reacción y reflejos en los estudiantes de la carrera de Ingeniería Automotriz en la Universidad Politécnica Salesiana, Quito, ya que surge la necesidad de poseer una herramienta de evaluación precisa y segura de las habilidades cognitivas y motoras de los conductores, esta se convierte en una herramienta esencial para los futuros profesionales en el ámbito automotriz, lo que motiva la creación de esta herramienta innovadora. La metodología, se basó a un enfoque cualitativo, descriptivo y experimental, ya que se detalló todos los aspectos técnicos del hardware y software que se utilizó en el desarrollo del simulador. Se describe la simulación realista de situaciones de conducción y los procedimientos de prueba y recopilación de datos. Las pruebas con estudiantes de Ingeniería Automotriz proporcionarán datos valiosos para evaluar la efectividad del simulador. El simulador de conducción no solo busca evaluar las habilidades cognitivas y motoras, sino que también se enfoca en proporcionar una experiencia amigable y segura para quienes realizan estas pruebas. La comparación de los resultados obtenidos de las pruebas realizadas con el simulador ha revelado información de gran relevancia para el proyecto. Entre estos resultados se destacan la capacidad de los estudiantes para mantener la concentración y reaccionar ante eventos inesperados durante la simulación. Se evaluó las diversas habilidades que poseen los estudiantes de la Universidad, a la vez sirvió para que practiquen y fortalezcan sus reflejos, ayudando a que estos se vayan desarrollando de forma correcta, para que así se disminuya los accidentes viales. En donde se concluyó que el diseño y desarrollo del simulador es una idea innovadora para poder determinar el tiempo de reacción y la resistencia a la monotonía que tiene cada uno de los participantes, esto ayuda a desarrollar la atención, concentración y la reacción a los eventos que se susciten en un tiempo determinado para evitar un incidente.

Palabras clave: Simulador de conducción, Reacciones y reflejos, Maniobras de conducción
Diseño de software.

ABSTRACT

The objective of this research project is to build a Reaction and Reflex Testing Driving Simulator for students of the Automotive Engineering program at Universidad Politécnica Salesiana in Quito. The necessity arises for a precise and safe evaluation tool for drivers' cognitive and motor skills, making it an essential tool for future professionals in the automotive field, thus motivating the creation of this innovative tool. The methodology was based on a qualitative, descriptive, and experimental approach, detailing all the technical aspects of the hardware and software used in the development of the simulator. It describes the realistic simulation of driving situations and the procedures for testing and data collection. Testing with Automotive Engineering students will provide valuable data to evaluate the simulator's effectiveness. The driving simulator aims not only to evaluate cognitive and motor skills but also to provide a user-friendly and safe experience for those taking these tests. The comparison of the results obtained from tests conducted with the simulator has revealed significant information for the project. Among these results, the students' ability to maintain concentration and react to unexpected events during the simulation stands out. Various skills possessed by the university students were evaluated, while also serving to practice and strengthen their reflexes, helping them develop correctly to reduce traffic accidents. It was concluded that the design and development of the simulator is an innovative idea to determine each participant's reaction time and resistance to monotony. This helps develop attention, concentration, and reaction to events occurring within a specific time to avoid an incident.

Keywords: Driving simulator, Reactions and reflexes, Driving maneuvers Software design.

INTRODUCCIÓN

Hoy en día, la seguridad vial se ha convertido en una preocupación global, dado que los accidentes de tráfico representan una de las principales causas de mortalidad y lesiones graves. Estos eventos afectan a diversas personas, en donde se destaca la urgencia de abordar de manera efectiva las habilidades de conducción para mitigar los riesgos asociados. A pesar de que el proceso para obtener una licencia de conducir incluye pruebas teóricas y prácticas, la evaluación de habilidades cognitivas y motoras necesarias para una conducción segura a menudo queda incompleta.

En este contexto, surge la necesidad de mejorar los métodos de evaluación para garantizar que los estudiantes posean las habilidades esenciales para enfrentar los desafíos de la conducción. La investigación se justifica por la insuficiencia de las pruebas convencionales para evaluar con precisión aspectos cruciales como la reacción simple que son elementos fundamentales para evitar situaciones peligrosas en la carretera. Ante esta brecha, la propuesta de desarrollar un simulador de conducción basado en tecnología de realidad virtual y sensores de movimiento se presenta como una solución innovadora y necesaria.

La presente investigación, se justifica en la creación y evaluación de un simulador de conducción con el objetivo de abordar desafíos específicos relacionados con la formación y entrenamiento de conductores. En donde la conducción segura y competente es esencial para la seguridad vial y la reducción de accidentes. Sin embargo, la formación convencional puede presentar limitaciones en términos de acceso a situaciones de conducción variadas, así como en la capacidad para exponer a los conductores a escenarios de riesgo controlados. La investigación abordará la necesidad de personalización en la formación de conductores, reconociendo que cada individuo tiene un nivel único de habilidades y experiencias previas. El simulador permitirá adaptar el programa de entrenamiento según las habilidades específicas de cada conductor, optimizando así el proceso de aprendizaje y mejorando la capacidad de reacción.

El proceso habitual para la obtención de una licencia de conducir suele basarse en pruebas teóricas y prácticas. Sin embargo, estas pruebas no siempre son suficientes para evaluar adecuadamente las habilidades cognitivas y motoras necesarias para una conducción segura. En particular, la reacción simple y la resistencia a la monotonía son aspectos importantes que pueden afectar la capacidad del conductor para evitar situaciones peligrosas en la carretera. Por ello, en este trabajo se propone desarrollar un simulador de conducción que

permita una evaluación precisa y objetiva de las reacciones simples y la resistencia a la monotonía de los solicitantes del permiso de conducción. Este simulador está diseñado para complementar la prueba tradicional y brindar una herramienta adicional para garantizar que los conductores estén en forma y listos para enfrentar los desafíos de conducir en la carretera.

Una respuesta simple se refiere a la capacidad de una persona para responder rápidamente a un estímulo externo. En el contexto de conducción, una respuesta simple y eficaz puede ayudar a prevenir o provocar un accidente. Por ello, es de suma importancia evaluar la capacidad de los futuros conductores para reaccionar rápidamente ante situaciones inesperadas en la carretera, como frenazos o giros bruscos.

Esto incluye el uso adecuado del volante y los controles auxiliares, como las luces direccionales, los faros y los limpiaparabrisas. La distracción puede dar lugar a movimientos imprecisos o lentos, lo que puede dar lugar a situaciones peligrosas en la carretera. El simulador de conducción propuesto en esta tesis estará basado en tecnología de realidad virtual y sensores de movimiento para reproducir una experiencia de conducción realista y controlada. El sistema mide y registra con precisión las reacciones simples de los participantes y la resistencia a la monotonía a través de una serie de pruebas diseñadas específicamente para evaluar estas habilidades.

Se espera que la introducción de este simulador de conducción como herramienta adicional en la tramitación del permiso de conducción reduzca el número de accidentes y siniestros provocados por conductores no cualificados. Además, se podrá identificar a las personas que necesitan formación adicional antes de obtener los permisos, lo que contribuirá a mejorar la seguridad vial y proteger la vida e integridad de los usuarios de la vía.

Esta investigación aborda las siguientes preguntas fundamentales: ¿Cómo pueden mejorarse las evaluaciones de habilidades de conducción para garantizar una reacción óptima y segura en los conductores? ¿En qué medida puede un simulador de conducción basado en realidad virtual proporcionar una evaluación precisa de las reacciones simples? Así mismo, se plantea la hipótesis de la introducción de este simulador como una herramienta que ayudará en el proceso de la obtención de una licencia, no solo contribuirá a la identificación de conductores más capacitados, sino que también reducirá significativamente los accidentes asociados con conductores no cualificados. El proyecto validará esta hipótesis a través de la aplicación de pruebas específicas y el análisis meticuloso de los resultados obtenidos.

En última instancia, se espera que esta investigación aporte no solo a la mejora de la seguridad vial, sino también al avance de las prácticas de evaluación y formación en el ámbito de la conducción.

PROBLEMA

Según el Instituto Nacional de Estadística y Censos (INEC), en el primer trimestre del 2023 en el Ecuador ocurrieron 4.991 siniestros de tránsito, una disminución del 2,9% con respecto al 2022 en el mismo periodo.

Analizando a mayor detalle esta estadística, en promedio se reportaron 1.664 siniestros mensuales de enero a marzo. En el primer trimestre, el 64,2% los siniestros de tránsito ocurrieron en el área urbana, mientras que en el área rural representaron el 35,8%.

La principal causa de siniestros, durante el primer trimestre, fue la impericia e imprudencia del conductor, con un total de 2.017 siniestros, que representan el 40,4%; seguido de no respetar las señales de tránsito con 1.113 siniestros, con el 22,3% del total de siniestros en el periodo.

En el trimestre de enero a marzo, los choques fueron los principales con un total de 2290 siniestros, el 45,9 %; seguido de atropellados con 705 siniestros, con el 14,1 % del total en el periodo.

En el primer trimestre hubo 5.098 víctimas, de las que 4545 fueron lesionados y 553 fallecidos, de los cuales el 89 % y 11 %, respectivamente.

Así también en el año 2023, se reportaron 5.098 víctimas, entre ellos 3.697 hombres que representan el 72,5% y 1.364 mujeres que representan el 26,8%.

El mayor número de víctimas se encuentran en el grupo de 30 a 45 años que representa el 19,81%, seguido de las personas de 18 a 29 años con 19,75%.

El alcohol es una de las principales causas del aumento de Siniestros de tránsito ya que este puede afectar la capacidad de un conductor para tomar decisiones correctas y seguras en la carretera y ralentizar el tiempo de reacción de un conductor, dificultando el tiempo de respuesta ante situaciones inesperadas. (ANT, 2021)

Este proyecto en la carrera de Ingeniería Automotriz ayudaría a reducir los índices ya mencionados anteriormente ayudando a mejorar la educación vial al proporcionar un espacio seguro para que los estudiantes practiquen sus habilidades mediante una experimentación segura.

En Ecuador, es evidente la ocurrencia cotidiana de siniestros en las carreteras, las cuales se caracterizan como accidentes de tránsito violentos. Se evidencia una tendencia a la disminución en los últimos cinco años, no obstante, contrastando con un marcado aumento en el año 2023, según reportes de la Agencia Nacional de Tránsito (ANT). En el primer semestre de dicho año, se registraron 1.156 víctimas, lo cual representa una preocupante alza en comparación con periodos precedentes.(Diario Primicias, 2023a).

En este escenario, el proceso de obtención de licencia de conducir se rige como una fase crítica para evaluar la aptitud de los conductores. A pesar de las pruebas teóricas y prácticas existentes, la falta de evaluaciones exhaustivas de habilidades clave como la reacción simple y la resistencia a la monotonía ha generado un vacío significativo en la preparación de conductores para afrontar situaciones reales en las carreteras.

La presente investigación se centra en abordar este desafío mediante la propuesta de un simulador de conducción innovador. Este simulador no solo busca complementar las pruebas convencionales, sino también brindar una evaluación precisa y objetiva de las habilidades cruciales para una conducción segura. La reacción simple, vital en momentos críticos, y la resistencia a la monotonía, esencial para un manejo preciso y seguro, serán objeto de evaluación detallada a través de la implementación de tecnologías avanzadas como la realidad virtual y sensores de movimiento.

Según el (Servicio Integrado de Seguridad, 2023) en Quito se registra 1400 accidentes de tránsito cada mes lo que indica que hay una clara falencia en las evaluaciones de los conductores del país. Con este proyecto se pronostica un avance significativo para la seguridad vial en Ecuador, por lo cual la implementación de un simulador de conducción que evalúa estas habilidades específicas contribuirá a la formación de conductores más competentes, a la vez que reducirá la incidencia de accidentes causados por deficiencias en estas áreas críticas. La aplicación directa de estos resultados se traducirá en vías más seguras, conductores mejor preparados y, en última instancia, en la protección de la vida y bienestar de todos los usuarios de las carreteras.

Con el respaldo de estadísticas oficiales y la conciencia creciente sobre la importancia de mejorar la seguridad vial, esta investigación no solo aborda un problema actual, sino que también propone una solución innovadora y práctica para fortalecer la preparación de conductores en Ecuador.

Para abordar eficazmente la problemática de seguridad vial en el país, es fundamental que el test de reacción simple sea incorporado de manera integral en las pruebas actuales. Este tipo de evaluación juega un papel crucial en la preparación para responder a situaciones de emergencia en las vías. Sin embargo, hasta el momento, su inclusión y evaluación no han sido abordadas de manera exhaustiva en los procesos de evaluación existentes. Esto deja a los conductores potencialmente mal preparados para lidiar con eventos repentinos, resultando en un aumento de accidentes de tráfico, de esta manera los estudiantes de Ingeniería Automotriz perfeccionaran su manejo y con ello la eficaz obtención de una licencia de conducir.

Las deficiencias en la evaluación de estas habilidades críticas impactan directamente en la seguridad vial. Conductores no aptos para reaccionar rápidamente pueden convertirse en un riesgo para sí mismos y para otros usuarios de las vías. La incidencia creciente de accidentes y lesiones relacionadas con el tráfico refuerza la necesidad de abordar estas limitaciones de manera efectiva.

En este contexto, el presente proyecto surge como respuesta a estas limitaciones identificadas en la evaluación de habilidades de conducción. El desarrollo de un simulador de conducción basado en tecnologías avanzadas busca superar las deficiencias actuales y ofrecer una solución práctica y efectiva para fortalecer la preparación de conductores y, en última instancia, mejorar la seguridad vial en Ecuador.

Este proyecto se justifica en el ámbito académico al ofrecer una herramienta innovadora que complementa la formación de los estudiantes de Ingeniería Automotriz de la Universidad Politécnica Salesiana. La implementación de un simulador de conducción proporcionará a los estudiantes una experiencia práctica y realista, enriqueciendo su aprendizaje y preparándolos para enfrentar los desafíos del mundo automotriz de manera más integral. Además, les permitirá aplicar los conocimientos teóricos adquiridos en un entorno virtual que simula situaciones de conducción reales, mejorando así su capacidad para abordar los problemas y las demandas del sector automotriz.

Desde una perspectiva social, la importancia de este proyecto radica en su contribución a la seguridad vial en la comunidad ecuatoriana. Al mejorar la evaluación de habilidades críticas de conducción, se busca reducir la incidencia de accidentes de tráfico, lesiones y pérdidas de vidas. La implementación de un simulador de conducción que evalúe la reacción simple no solo beneficiará a los estudiantes de ingeniería automotriz, sino que también en un futuro podrá repercutir directamente en la seguridad de todos los usuarios de las carreteras, creando un impacto positivo y duradero en la sociedad, permitiendo que el usuario pueda obtener habilidades efectivas al momento de reaccionar.

Desde el punto de vista económico, este proyecto tiene el potencial de generar beneficios significativos al reducir los costos asociados con accidentes de tránsito. La mejora en la formación de conductores y la disminución de incidentes en las carreteras se traducirán en ahorros considerables en términos de gastos médicos, reparaciones de vehículos y pérdidas de productividad. Además, al formar conductores más competentes, se podría anticipar una disminución en la prima de seguros, generando un impacto positivo en la economía tanto a nivel individual como a nivel nacional.

Delimitación del problema.-

El propósito de esta tesis consiste en abordar de manera rigurosa el tema de la evaluación de la reacción simple dentro del proceso de obtención de licencias de conducir, particularmente enfocado en los estudiantes de la carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana en Quito. El simulador de conducción propuesto para esta investigación será diseñado como una herramienta integral que reproduzca de manera realista situaciones de conducción. El simulador, basado en tecnología de realidad virtual y sensores de movimiento, permitirá evaluar de manera precisa las habilidades de reacción y reflejos de los estudiantes de Ingeniería Automotriz. Que utiliza un entorno de conducción con la señalética de Pare, implementando respuestas rápidas a eventos imprevistos, como frenazos bruscos midiendo así el tiempo de reacción.

El entorno de evaluación de este proyecto se establecerá dentro de la carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana en Quito, Ecuador. Con la finalidad de contribuir al desarrollo académico y práctico de los estudiantes, se propone la donación del simulador de conducción que tiene como objetivo enriquecer las experiencias de aprendizaje

en las materias relacionadas con la formación automotriz y fortalecer el laboratorio de la carrera.

Tiempos de reacción:

El tiempo de reacción establecido para este proyecto, conforme a los estándares requeridos por la Agencia Nacional de Tránsito (ANT), es de 0.43 segundos. En caso de no cumplir con este parámetro, el resultado final del test de evaluación de reacción será catalogado como "reprobado". (Bima, 2023)

La investigación y desarrollo del simulador de conducción se destinará como una donación a la carrera de Ingeniería Automotriz, para su integración en los laboratorios correspondientes. Sin embargo, a pesar de los beneficios y avances que el simulador de conducción pueda ofrecer, es crucial identificar y abordar posibles limitaciones que podrían surgir durante su implementación y uso, tales como:

- Aunque el simulador se esfuerza por proporcionar una experiencia de conducción realista, es factible que ciertos aspectos, como las sensaciones físicas y la percepción del entorno, no logren alcanzar un nivel de realismo completo.
- Existen posibles limitaciones en la precisión de los datos recopilados, especialmente en lo que respecta a la medición de respuestas fisiológicas y emocionales del conductor.
- La implementación exitosa del simulador puede depender en gran medida de la disponibilidad y el mantenimiento de hardware y software específicos, lo que podría generar desafíos logísticos y financieros.
- A pesar de la autonomía que puede proporcionar el simulador, es probable que la supervisión docente constante sea esencial para garantizar un uso efectivo y seguro del mismo.

Objetivo General.

Desarrollar un simulador de conducción a través del programa Visual Studio para la Evaluación de reacciones y reflejos en estudiantes de Ingeniería Automotriz, Sede Quito.

Objetivos Específicos.

- Realizar un levantamiento de información para identificar y comprender las leyes y reglamentos vigentes en Ecuador que rigen el desarrollo y uso de simuladores.
- Implementar un simulador de conducción utilizando el programa Visual Studio.
- Evaluar la utilidad y eficacia del simulador de conducción mediante pruebas de diagnóstico y funcionamiento real.

Marco Teórico

La Seguridad vial, es un tema de gran importancia a nivel mundial. Los accidentes de tráfico son una de las causas más comunes de muerte y lesiones en todo el mundo. Las habilidades de conducción son un factor importante para la seguridad vial. Los conductores con buenas habilidades de conducción son menos propensos a sufrir accidentes.

Las evaluaciones de conducción pueden ayudar a identificar a los conductores que necesitan formación para mejorar la seguridad vial. Los simuladores de conducción son una herramienta valiosa para evaluar las habilidades de conducción. Estos dispositivos recrean virtualmente escenarios de conducción realistas, lo que permite a los conductores practicar y dominar diversas habilidades en un entorno seguro y controlado.

Este marco teórico examina los conceptos fundamentales de los simuladores de conducción, su aplicación a la seguridad vial y su potencial para reducir los accidentes.

Los Conceptos básicos de los simuladores de conducción son sistemas diseñados para ofrecer a los usuarios una experiencia de conducción inmersiva y controlada.

En líneas generales, el modelo de vehículo constituye una representación virtual del vehículo real, abarcando detalles sobre sus características físicas y dinámicas. El sistema de visualización posibilita a los conductores tener una vista realista del entorno de conducción, mientras que los controles de conducción les facultan para llevar a cabo acciones y maniobras análogas a las que realizarían en un vehículo real.

Además, los simuladores de conducción tienen el potencial de reducir los accidentes de tránsito de varias maneras. En primer lugar, pueden ayudar a identificar a los conductores con necesidades de capacitación, lo que puede conducir a una mejora de las habilidades de

conducción, asimismo puede ayudar a mejorar la seguridad en las carreteras. En tercer lugar, los simuladores de conducción pueden utilizarse para investigar accidentes, lo que puede ayudar a desarrollar medidas de seguridad más eficaces (Osorio & Vera, 2021).

Los simuladores de conducción son herramientas valiosas para evaluar habilidades y reducir accidentes, y ofrecen grandes beneficios. Permiten practicar en un entorno seguro y pueden utilizarse en diversas áreas de la seguridad vial, desde el desarrollo de vehículos hasta la formación de conductores y la investigación de accidentes. Hay muchas posibilidades para reducir los accidentes. Además, también se podrán resolver cuestiones como el coste y la disponibilidad de estos simuladores y su aplicación en la formación de conductores y en la rehabilitación de conductores discapacitados, ampliando aún más el alcance y la utilidad de los simuladores en diversas situaciones.

2.1 Evaluación de Habilidades en la Conducción

Las habilidades que se evalúan en los conductores son las cognitivas y motoras, las cuales desempeñan un papel fundamental en la conducción segura. Cuando se refiere a las habilidades cognitivas, se habla de los procesos mentales como la atención, percepción, memoria, toma de decisiones y la resolución de problemas. Por otro lado, las habilidades motoras se refieren a la coordinación física y la destreza necesarias para controlar el vehículo de manera efectiva. Ambos tipos de habilidades son indispensables para realizar una conducción segura y eficiente.

2.1.1 Habilidades cognitivas

Atención: Es una habilidad cognitiva clave en la conducción. Los conductores deben mantener la atención constante en el entorno vial, prestando atención a los vehículos que los rodean, las señales de tráfico, los peatones y otros factores relevantes. La falta de atención o la distracción pueden aumentar significativamente el riesgo de accidentes. La capacidad de dividir la atención entre diferentes estímulos y mantener la concentración en la tarea de conducción es esencial para tomar decisiones informadas y reaccionar rápidamente ante situaciones imprevistas (García et al., 2019).

Percepción: También juega un papel crucial en la conducción, los conductores deben ser capaces de percibir adecuadamente la información visual y auditiva que reciben del entorno vial. Esto implica reconocer y comprender las señales de tráfico, interpretar las acciones de otros conductores y anticipar posibles peligros. Una percepción deficiente puede llevar a una respuesta tardía o incorrecta, lo que aumenta el riesgo de accidentes.

Memoria: Es importante en la conducción, especialmente la memoria a corto plazo. Los conductores deben ser capaces de recordar y aplicar las reglas de tráfico, así como recordar información relevante sobre el camino, como la ubicación de los cruces o los límites de velocidad. Además, la memoria a largo plazo desempeña un papel en la formación de hábitos y conductas seguras. Los conductores experimentados confían en su memoria para realizar maniobras básicas de manera automática y eficiente.

Toma de decisiones y la resolución de problemas: Son habilidades cognitivas esenciales en la conducción. Los conductores deben ser capaces de evaluar rápidamente las situaciones, analizar los diferentes cursos de acción disponibles y elegir la opción más segura y adecuada. Esto implica evaluar los riesgos, anticipar las consecuencias de las decisiones y ajustar el comportamiento en función de las condiciones cambiantes. La capacidad de tomar decisiones informadas y reaccionar de manera adecuada en situaciones de emergencia es fundamental para evitar accidentes.

2.1.2 Habilidades motoras

En cuanto a las habilidades motoras, la coordinación física y la destreza son indispensables para controlar el vehículo de manera segura y precisa. Esto incluye habilidades como el uso adecuado de los pedales, el volante y la palanca de cambios, así como la capacidad de realizar maniobras como el estacionamiento, los cambios de carril y las vueltas de manera suave y controlada. La coordinación ojo-mano y el tiempo de reacción también son importantes para realizar maniobras defensivas y evitar colisiones (Analuisa, 2019).

Por lo antes mencionado, se evidencia que las habilidades cognitivas y motoras son fundamentales en la conducción segura. Los conductores deben contar con una atención adecuada, una percepción precisa, una memoria funcional, una capacidad de toma de decisiones sólida y habilidades motoras bien desarrolladas. El desarrollo y mantenimiento de estas habilidades son responsabilidad de los conductores, quienes pueden mejorarlas a través de la educación vial, la práctica constante y la conciencia de la importancia de estar en óptimas condiciones cognitivas y físicas al volante. Al hacerlo, se promueve la seguridad vial y se reducen los riesgos de accidentes.

2.1.3 Métodos y herramientas convencionales utilizados para evaluar habilidades de conducción.

Test de resistencia a la monotonía y de reacción simple

El test de resistencia a la monotonía y la prueba de reacción simple son dos herramientas utilizadas para evaluar las habilidades motoras y cognitivas de una persona. Estas pruebas son comúnmente utilizadas en el campo de la psicología y la medicina para medir la función cerebral, la coordinación y la atención.

El test de resistencia a la monotonía evalúa la capacidad de una persona para concentrarse al momento de conducir, valorando la habilidad que tiene el usuario al momento de circular por una vía larga sin perder la concentración. La prueba generalmente implica el uso de una tarea en la que se le pide al individuo que realice movimientos sincronizados o alternados con ambas manos. Esto puede incluir tareas como tocar puntos específicos en un tablero con ambas manos simultáneamente, o seguir una secuencia de movimientos predeterminada (Vera, 2019).

El objetivo de esta prueba es evaluar la capacidad de una persona para coordinar los movimientos de ambas manos de manera precisa y eficiente. Esto puede proporcionar información sobre el estado de la función motora fina y gruesa, así como la integración sensoriomotora en el cerebro. Un rendimiento deficiente en esta prueba puede sugerir problemas de coordinación, lesiones cerebrales o trastornos neuromusculares.

Por otro lado, la prueba de reacción simple evalúa la velocidad de respuesta de una persona ante un estímulo específico. En esta prueba, el individuo se le pide que reaccione lo más rápido posible cuando se presenta un estímulo, como una luz o un sonido. Por lo general, se mide el tiempo transcurrido entre la presentación del estímulo y la respuesta del individuo (RAUCH, 2019).

La prueba de reacción simple se utiliza para evaluar la velocidad de procesamiento cognitivo y la capacidad de respuesta rápida. También puede proporcionar información sobre la atención y la capacidad de concentración de una persona. Un tiempo de reacción más rápido generalmente se considera indicativo de una mejor función cognitiva y un sistema nervioso más ágil.

Estas pruebas se realizan en entornos controlados y suelen ser administradas por profesionales capacitados, como psicólogos o médicos. Los resultados se registran y analizan para obtener una medida objetiva de las habilidades motoras y cognitivas de una persona.

Además de su aplicación clínica, estas pruebas también se utilizan en la investigación científica para estudiar el desarrollo motor, la plasticidad cerebral y los efectos de diferentes

intervenciones terapéuticas. Los datos recopilados de estas pruebas pueden proporcionar información valiosa para comprender los mecanismos subyacentes de la coordinación motora y el procesamiento cognitivo.

La prueba de resistencia a la monotonía y la prueba de reacción simple son dos herramientas importantes para evaluar las habilidades motoras y cognitivas de una persona. Estas pruebas proporcionan información objetiva sobre la coordinación motora, la velocidad de procesamiento cognitivo y la capacidad de respuesta rápida. Tanto en el ámbito clínico como en la investigación científica, estas pruebas desempeñan un papel crucial en la evaluación y comprensión de las capacidades humanas en términos de coordinación y atención

2.2 Tecnología y Simuladores en la Conducción

Los simuladores de conducción son dispositivos tecnológicos avanzados diseñados para recrear la experiencia de conducir un vehículo en un entorno controlado. Estos sistemas pueden ser utilizados para una amplia variedad de propósitos, que van desde la formación de conductores hasta la realización de investigaciones científicas. Permiten a los usuarios experimentar de manera realista diferentes situaciones de manejo, ofreciendo un entorno seguro y controlado para practicar habilidades de conducción o llevar a cabo estudios en condiciones controladas.

El uso más común de los simuladores se evidencia en el ámbito de la formación de conductores, donde desempeñan un papel fundamental. Estos sistemas permiten enseñar a los conductores una amplia gama de habilidades necesarias para operar vehículos en entornos controlados y seguros, antes de enfrentarse a las condiciones reales de la carretera. Además de la instrucción para conductores civiles, los simuladores también se utilizan en el entrenamiento de operadores de vehículos comerciales, como camiones y autobuses. Incluso en el ámbito militar, estos simuladores ofrecen un entrenamiento avanzado para conductores, contribuyendo así a la preparación y seguridad en diversas áreas de conducción.

(Herrero, 2020).

Los simuladores de conducción se emplean para realizar estudios sobre la conducta humana en situaciones de tráfico vehicular, evaluar el impacto del diseño de carreteras y señalizaciones en la seguridad vial, así como también para investigar el impacto de los vehículos autónomos. La creación de estos simuladores requiere la integración de tecnología avanzada, incluyendo software de simulación, hardware informático, dispositivos de entrada y sensores. El software de simulación se utiliza para desarrollar modelos virtuales de

entornos de conducción, haciendo uso de tecnologías de inteligencia artificial. Por su parte, el hardware se encarga de ejecutar el software y procesar los datos de entrada. En algunos casos, estos simuladores incluyen un "cockpit" o cabina de simulación que imita a las cabinas de los vehículos reales, junto con pantallas o sistemas de proyección que proporcionan representaciones visuales de los entornos de conducción (García, 2019).

También se necesita dispositivos de entrada, como son los volantes, pedales y los cambios de las marchas, que se usan para la simulación de las experiencias de la conducción, como que fuera un vehículo real, posee sensores, cámaras y sensores láser, se usan para la medición de las posiciones y los movimientos de los simuladores y proporcionar una experiencia que sea más realista.

Por otro lado, se puede decir que la tecnología de simulación en la actualidad es fundamental para el mejoramiento de las destrezas al volante, este ayuda a la vez el desarrollo de la confianza de los conductores que recién están empezando con sus actividades, antes de que se enfrenten al tráfico real. Los distintos instructores de las escuelas han notado una disminución en los tiempos necesarios para que los alumnos vayan adquiriendo diferentes habilidades de manejo que estos sean sólidos, lo que va contribuyendo a las formaciones más seguras y eficientes.

2.3 Factores de Riesgo y Seguridad Vial

2.3.1 Seguridad vial

Son las situaciones por las que en las vías no se encuentran demasiados riesgos o daños a causa de la movilidad de los automóviles, por ende, se puede notar una serie de sistemas o normas con las que se van disminuyendo la posibilidad de encontrarse con algunos choques y averías, lo que provoca muchas consecuencias. Su propósito es la protección de los bienes y personas, por medio de la erradicación o del control del factor de riesgo que ayudan a la reducción de siniestros de tránsito.

En el Ecuador se puede observar que todos los días existen muertes en las vías, los cuales se revelan como accidentes violentos de tránsito. A lo largo de los años, se han observado una reducción constante en estos valores, Sin embargo, en el año 2023 se registró un aumento significativo, según la Agencia Nacional de Tránsito (ANT), en los primeros seis meses se reportaron 1.156 víctimas (Diario Primicias, 2023b).

Por lo antes mencionado, se debe dar a conocer los tiempos de reacciones que poseen los conductores, en donde se puede decir según (Jerez et al., 2020) menciona que el tiempo de reacción en la conducción “se basa al periodo de tiempo que pasa desde que un conductor percibe un obstáculo o situación de emergencia hasta que reacciona”. Al conocer esta definición, se marca la diferencia que existe entre sufrir un accidente o manejar la situación y así poder salvar las dificultades que se presentan, en donde es fundamental el conocimiento de los diversos elementos que intervienen para evitarlos.

Por lo cual, es imprescindible practicar con un simulador, ya que este tiene el propósito de eliminar los distintos vicios o mañas al momento de conducir, al mismo tiempo que mejora las capacidades de respuesta ante una determinada situación o también comprobar las reacciones ante ciertos contratiempos. Por todo esto que se menciona es fundamental que los conductores de carga pesada lo practiquen, ya que ellos pasan mucho tiempo en el volante y van adquiriendo malos hábitos, los que se deberían corregir, esto aplica también en las personas que se dedican al taxismo, cambiando estos problemas y asegurando una excelente seguridad vial.

2.3.2 Factores de riesgo

Entre los diversos factores de riesgo inherentes a la conducción, destacan principalmente tres: el conductor, el vehículo y el entorno vial.

El conductor, siendo el elemento más crucial, influye significativamente en la seguridad vial, ya que su habilidad, conocimiento y capacidad para evitar conductas riesgosas son determinantes. En segundo lugar, se encuentra el estado del vehículo, el cual debe mantenerse en óptimas condiciones, abarcando aspectos como el buen estado de las llantas, el correcto funcionamiento de los sistemas de dirección y frenos, así como la adecuada iluminación del vehículo. Por último, el entorno y las condiciones de la vía también ejercen un impacto considerable, considerando factores como las condiciones atmosféricas, el volumen de tráfico, y las características de la carretera, ya sea pavimentada o empedrada, junto con la disponibilidad de equipos de seguridad y una iluminación adecuada (Nhtsa, 2024)

También se puede encontrar los factores de riesgo que se relaciona con el conductor, los cuales se derivan al alcoholismo, las drogas, algunos medicamentos, la fatiga y a la vez del sueño, las cuales también se deriva de la distracción, las velocidades y también de algunos

factores psicológicos, por todas estas razones, es imprescindible poseer gestiones preventivas que se van presentando para disminuir los accidentes de tránsito.

2.4 Estudios Comparativos y Edad en la Conducción.

Se puede decir que el tiempo continuo de conducción de las personas influyen de manera directa en los riesgos de mostrar diversos síntomas como la fatiga, cansancio sea físico o emocional, por lo cual esto empuja a ser partícipe de un siniestro de tránsito en donde se evidencia lesiones a terceros o personales y en casos más trágicos hasta la muerte, por ende, se evidencia que la edad, interviene mucho en los reflejos, ya que si la edad es avanzada los movimientos y reflejos son más lentos.

También, perjudican las altas temperaturas, en donde las temperaturas altas tienden a adormecer a las personas, lo que incide de forma directa en la velocidad y los reflejos de sus respuestas. A la vez se presenta el estado emocional, lo que influyen de forma directa en los tiempos de reacciones de los conductores, por ejemplo, si se encuentran al volante pensando en los problemas o con mucho estrés, es posible que se encuentren menos concentrado y sin la capacidad para actuar frente a los peligros que se verán mermados.

En la actualidad, en Ecuador no existen leyes, normativas o resoluciones que se enfocan en los diversos tiempos de conducción para el transporte terrestre, este puede ser, comercial, publico, particular o por cuenta propia, afirmando esto después de verificar que en la Agencia Nacional de Tránsito no existen normas o regulaciones con reseña al tiempo de conducciones e informaciones que fueron constatadas (Andrade & Gonzales, 2020).

2.5 Normativas y Estándares en Conducción

A continuación, se detallan las normativas y estándares en la conducción. Para comenzar, se hace referencia al Registro Oficial Suplemento del año 2019, donde se encuentra lo siguiente:

Art. 4.- Es obligación del Estado garantizar el derecho de las personas a ser educadas y capacitadas en materia de tránsito y seguridad vial, en su propia lengua y ámbito cultural. Para el efecto, el Ministerio del Sector de la Educación en coordinación con la Agencia Nacional de Regulación y Control del Transporte Terrestre, Tránsito y Seguridad Vial, desarrollarán los programas educativos en temas relacionados con la prevención y seguridad vial, principios, disposiciones y normas fundamentales que regulan el tránsito, su señalización considerando la realidad lingüística de las

comunidades, pueblos y nacionalidades, el uso de las vías públicas, de los medios de transporte terrestre y dispondrán su implementación obligatoria en todos los establecimientos de educación, públicos y privados del país.

Se encuentra en el Reglamento a ley de transporte terrestre publicado en el año 2012 en donde se encuentra plasmado lo siguiente:

Art. 3.- El sistema de gestión de la Agencia Nacional de Tránsito de la Comisión de Tránsito del Ecuador se sustentará en un proceso continuo de planeamiento estratégico; de gestión por procesos; de medición, control de calidad y de sistemas de mejora continua. (Seguridad vial, 2012).

Parámetros de Respuesta test de resistencia a la monotonía

La normativa estandarizada del test de resistencia a la monotonía es crucial para garantizar la validez, confiabilidad y objetividad de la evaluación. Esta normativa se basa en principios psicométricos y estadísticos que incluyen:

Definición clara de la tarea y los estímulos: Especificar con precisión la tarea, los estímulos y sus características.

Instrucciones estandarizadas: Proporcionar instrucciones claras, concisas y uniformes a todos los participantes.

Parámetros de respuesta precisos: Definir claramente qué se espera que haga el participante ante cada estímulo.

La reacción ante un estímulo inmediato, el tiempo de respuesta de un conductor promedio oscila dentro de los 0.5 hasta 1 segundos. Este intervalo varía según diferentes factores que atañen directamente al conductor y nunca al estado de la carretera o cualquier otro motivo externo. (Bima 2023)

La norma ISO 39001, relativa a los Sistemas de Gestión de la Seguridad Vial, es fundamental para las organizaciones que buscan reducir, e incluso eliminar, la incidencia y el riesgo de muertes y lesiones graves derivadas de accidentes de tráfico. En este contexto, los simuladores de conducción emergen como un recurso complementario de gran valor para la implementación efectiva de dicha norma.

El Reglamento a la Ley de Tránsito y Transporte Terrestre complementa estas disposiciones, proporcionando directrices detalladas sobre los requisitos técnicos y pedagógicos que deben cumplir los simuladores utilizados en programas de formación vial. Estas regulaciones son fundamentales para asegurar la efectividad y calidad de los simuladores, así como para

promover una educación vial integral y orientada a la seguridad en Ecuador. (Patricio Castillo Villacrés Eugenio Javier Escobar Gonzáles y del Carmen Viteri Naranjo 2020)

2.6 Relevancia del Simulador Propuesto

Visual Studio es un instrumento que permite desarrollar de forma eficiente y eficaz los ciclos de desarrollo en cualquier lugar, permite el desarrollo integrado para que se pueda editar, escribir, compilar y depurar diversos código y ayudar a la implementación del simulador, con las ediciones y depuraciones de los distintos códigos, este programa incluye herramientas, compiladores de finalización, controles de códigos fuentes, extensión y muchas más opciones para el mejoramiento de las distintas fases de los procesos de desarrollo de software.

Este aspecto es de vital importancia para el correcto funcionamiento del simulador y su capacidad para asistir a los individuos en la mejora de sus reflejos y en la modificación de malos hábitos de conducción, especialmente aquellos conductores que operan vehículos diariamente en entornos laborales.

CAPÍTULO 1

DESARROLLO DEL PROYECTO

1.1 Software

Con el objetivo de desarrollar un programa que satisfaga la necesidad de crear un simulador de conducción, capaz de evaluar tanto la reacción a través del test de reacción simple como la capacidad de coordinación frente a ritmos y tiempos impuestos mediante el Test de Resistencia a la Monotonía, se ha optado por utilizar las herramientas disponibles en Visual Studio 2022. Esta elección se fundamenta en la versatilidad y potencialidad de este entorno de desarrollo para crear aplicaciones interactivas y funcionales, lo que permitirá la implementación efectiva de las pruebas mencionadas, proporcionando una herramienta eficaz para evaluar habilidades de conducción.

Se utilizan los programas Visual Studio, ya que ofrece una potente combinación de herramientas, recursos y flexibilidad para desarrollar un simulador de conducción, permitiendo una integración fluida entre hardware y software, los cuales ofrecen opciones de programación versátiles, herramientas de depuración, pruebas sólidas, la capacidad de personalizar y escalar el proyecto según sus necesidades.

Con el manejo del software Visual Studio se permitirá crear el programa combinando comandos y herramientas, los cuales, ayudando a la creación del proyecto facilitarán la interacción con los usuarios del simulador.

1.1.1. Visual Studio 2022

Visual Studio es un entorno de desarrollo integrado (IDE) ampliamente reconocido y utilizado en la creación de diversas aplicaciones de software. Su conjunto de características y herramientas proporciona una plataforma sólida para el desarrollo de un simulador de conducción destinado a medir la reacción mediante el Test de Reacción Simple, así como para evaluar la capacidad de coordinación frente a ritmos y tiempos impuestos a través del Test de Resistencia a la Monotonía. La versatilidad y eficacia de Visual Studio hacen que sea una elección idónea para este proyecto, facilitando la implementación de las funcionalidades necesarias y garantizando un desarrollo eficiente de alta calidad del simulador de conducción.

Es esencial tener en cuenta la programación que será utilizada durante la ejecución del proyecto. Estos programas son fundamentales para garantizar resultados satisfactorios que beneficien a los estudiantes de la Facultad de Ingeniería Automotriz de la Universidad

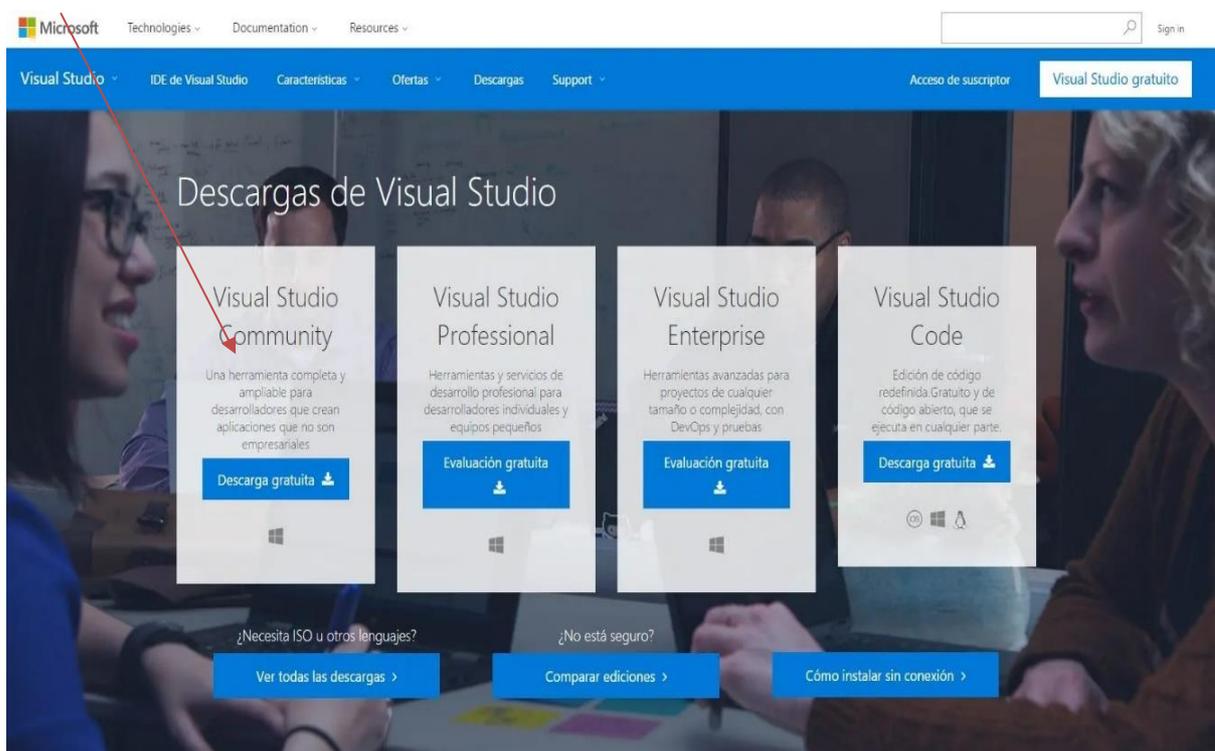
Politécnica Salesiana, así mismo puedan impartir su formación académica no solamente con teoría sino también con la práctica, respetando las normas y reglamentos de la educación vial.

Entre las características por la que se seleccionó el programa para desarrollar el simulador de conducción se encuentran detalladas a continuación mediante la representación del uso de cada una de ellas:

Fácil instalación del software

Se procede a ingresar al link para instalar inmediatamente con solo un clic, al siguiente enlace el cual permite la obtención del programa.

Figura 1.1: Descargas de Visual Studio



Fuente: (Microsoft, 2023a)

Figura 1.2: Bienvenidos al programa

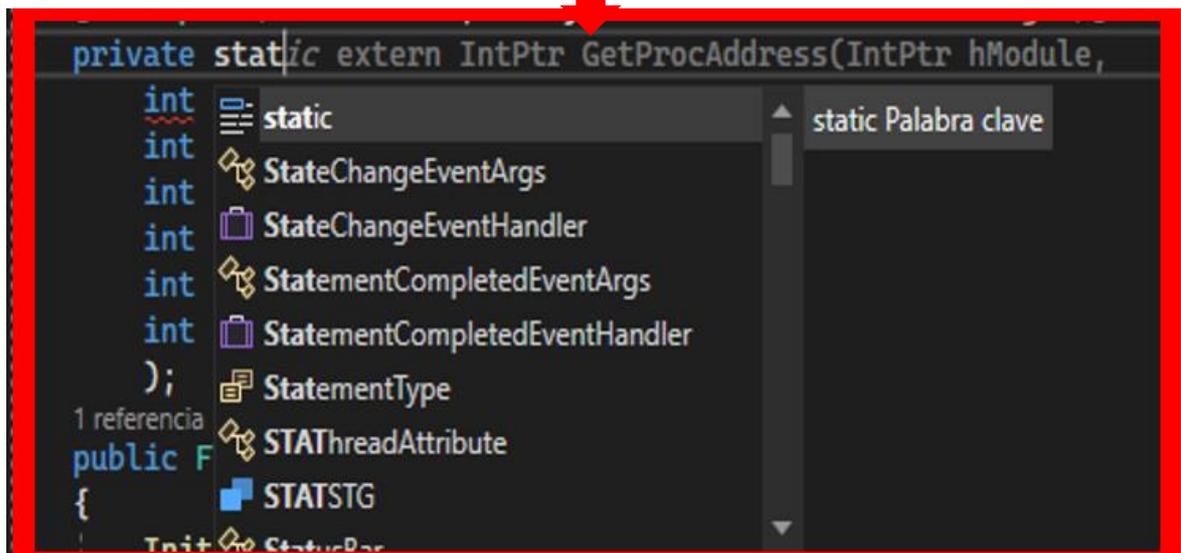
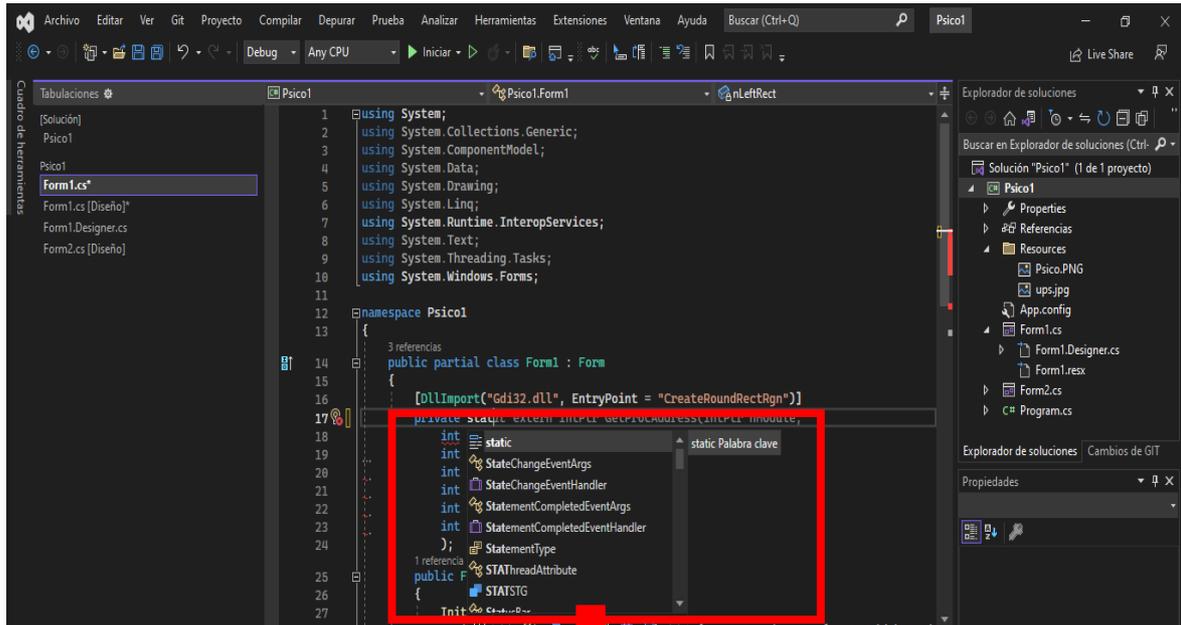


Fuente: (Microsoft, 2023b)

Edición de código

Visual Studio ofrece un editor de código potente y versátil con resaltado de sintaxis, autocompletado y sugerencias inteligentes. Esto facilita la escritura y edición del código fuente del simulador de conducción.

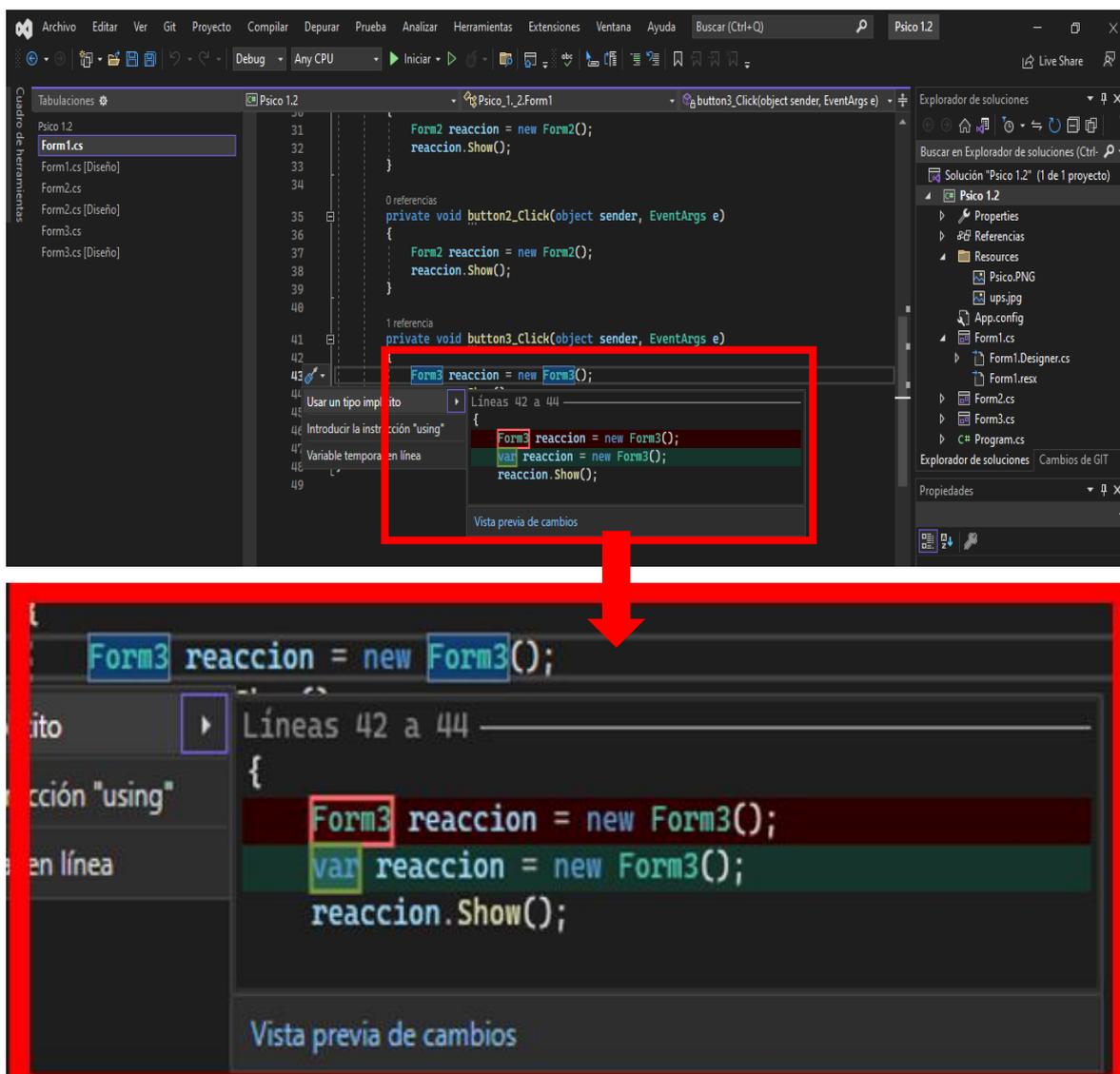
Figura 1.3: Visual Studio Ejemplo Autocompletado.



Depuración

El entorno de depuración de Visual Studio permite ejecutar el código paso a paso, establecer puntos de interrupción y examinar variables en tiempo de ejecución. Esto es especialmente útil para identificar y corregir errores en el simulador de conducción.

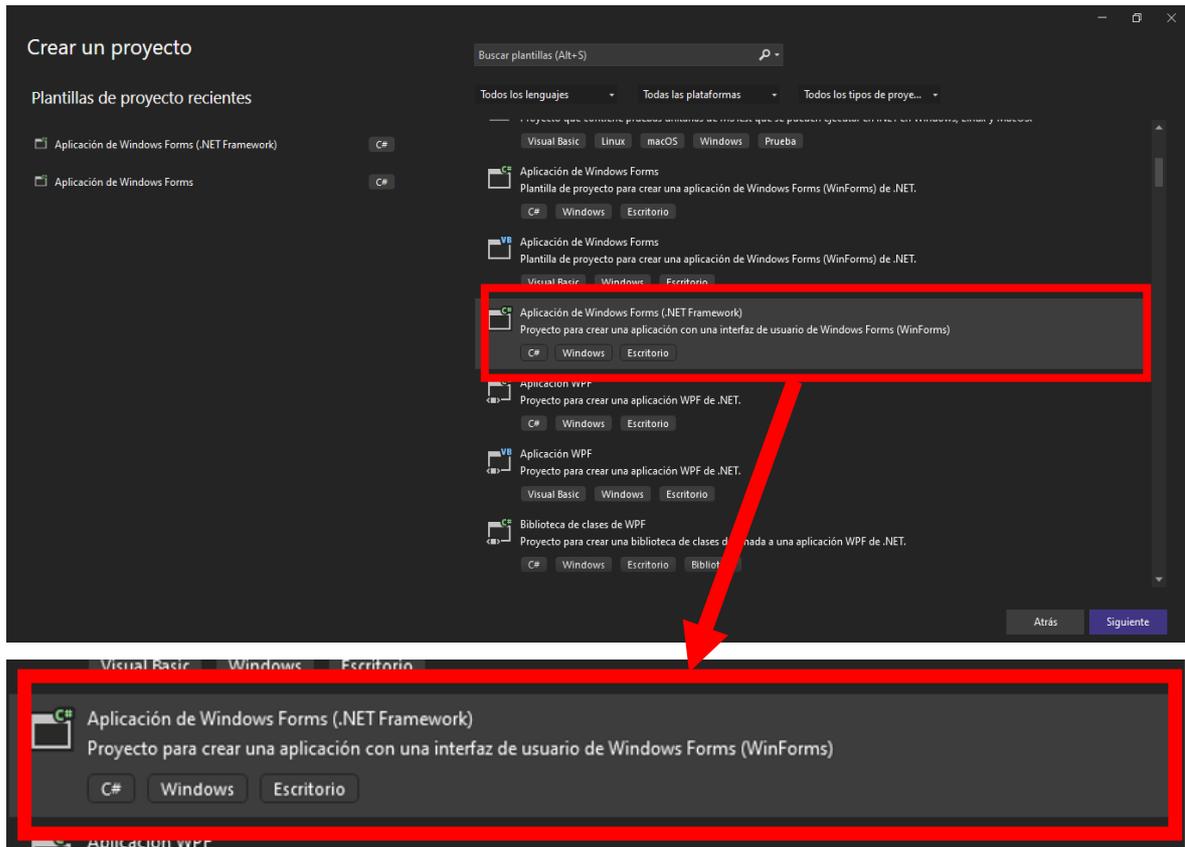
Figura 1.4: Visual Studio Corrección De Errores



1.1.2. Elaboración De Pantalla De Inicio con C#

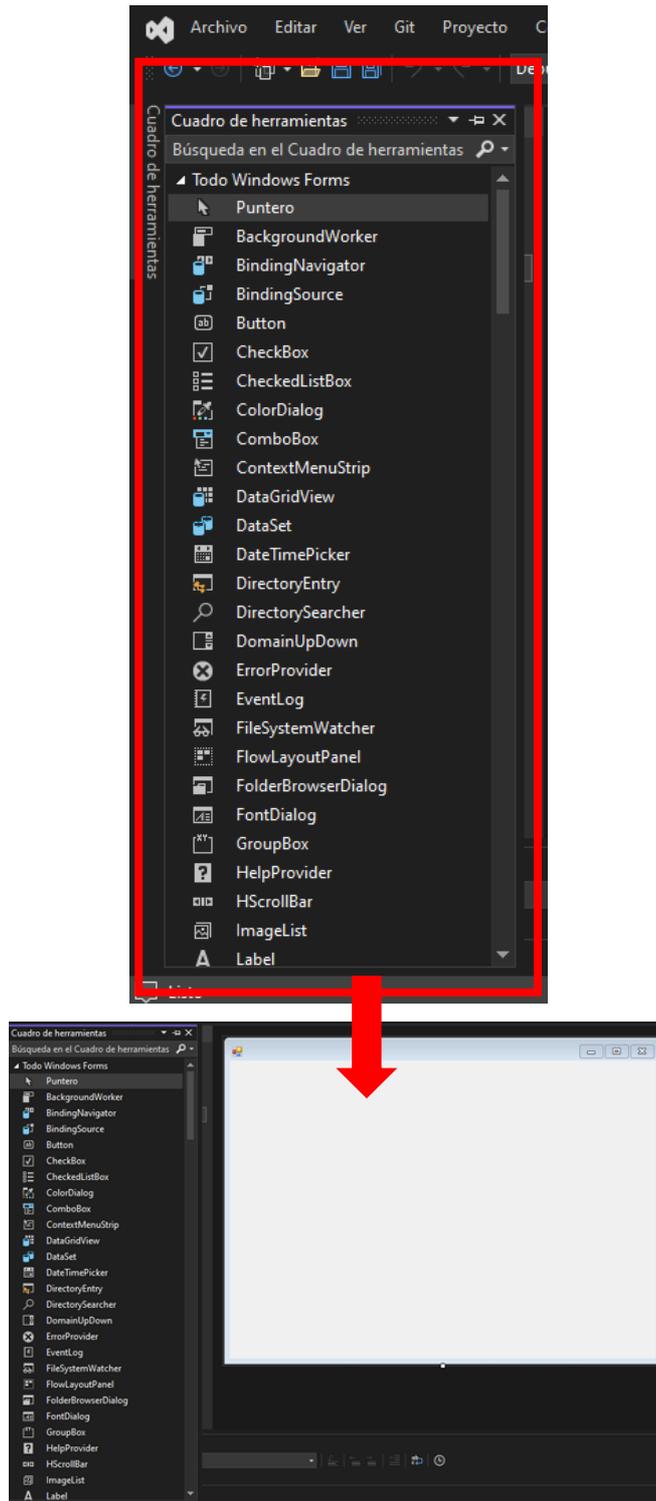
- Para empezar a realizar la pantalla de inicio en Visual Studio, es necesario buscar la opción más adecuada que permita adaptarse a la necesidad y caracteres requeridos por los desarrolladores, en este caso se accede a la opción Windows Form, que cuenta con una gran biblioteca para realizar una buena presentación de la pantalla de inicio del simulador.

Figura 1.5: Aplicación Windows Forms.



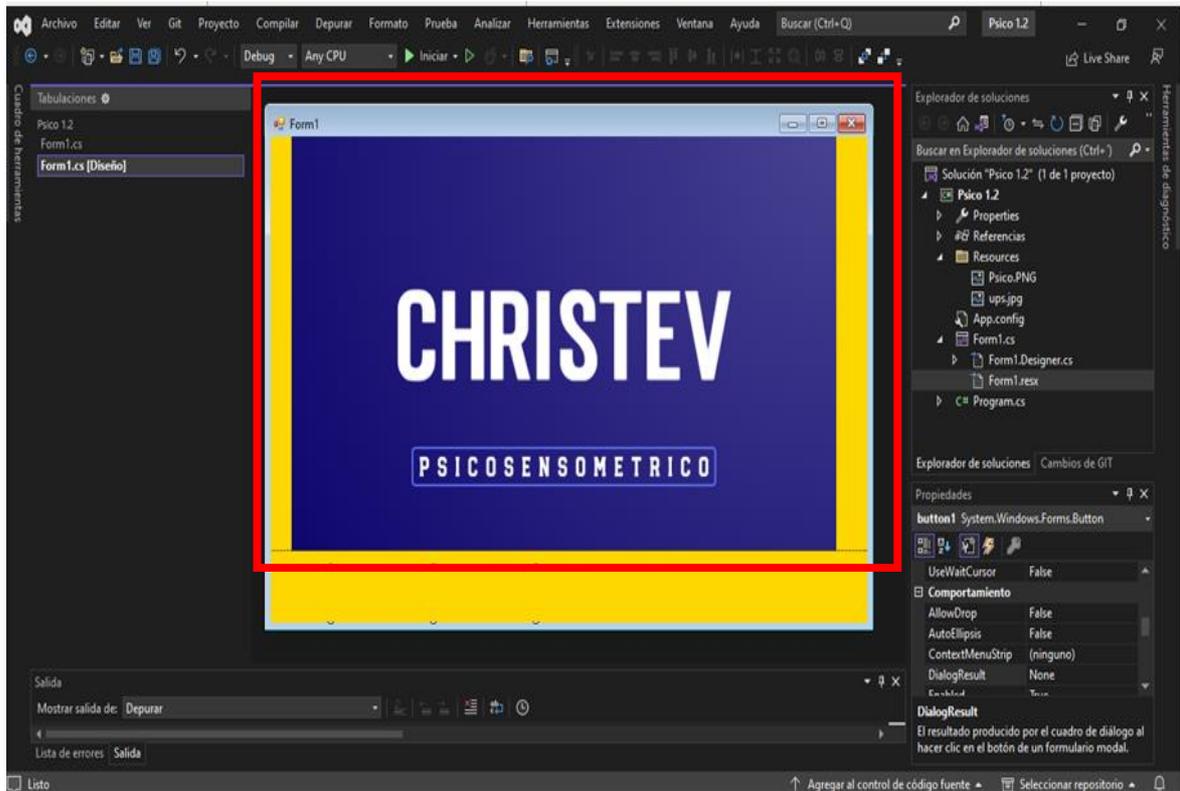
- Al seleccionar esta opción, se abre una pantalla en blanco y al lado izquierdo aparece el cuadro de herramientas, en donde se identifican algunas herramientas muy útiles como el de colocar un texto y una imagen a tu elección, con todo lo antes mencionado se puede crear la pantalla de inicio y darle la contextura al gusto del diseñador tomando en consideración todos los iconos que puede utilizar, para la utilización de cada herramienta lo único que se hace es seleccionar y arrastrarle hacia la pantalla en blanco.

Figura 1.6: Cuadro Herramientas.



- Una vez ubicada cada herramienta se podrá elaborar el boceto de pantalla de inicio, con el fin de dar una imagen adecuada e innovadora al simulador de conducción, usando todas las herramientas necesarias para elaborar el diseño de inicio de pantalla el cual se lo ha realizado con la siguiente representación:

Figura 1.7: Boceto pantalla de inicio.



- Una vez realizado el boceto de la pantalla de inicio, se procede a crear otro boceto de una pantalla en el cual ira el simulador de conducción, la cual permitirá medir los tiempos del test de reacción simple de cada uno de los usuarios a ser evaluados, pero antes se coloca un botón para poder redireccionar de la pantalla principal de inicio a la pantalla de Test de reacción simple.

Figura 1.8: Identificación del “Button”.

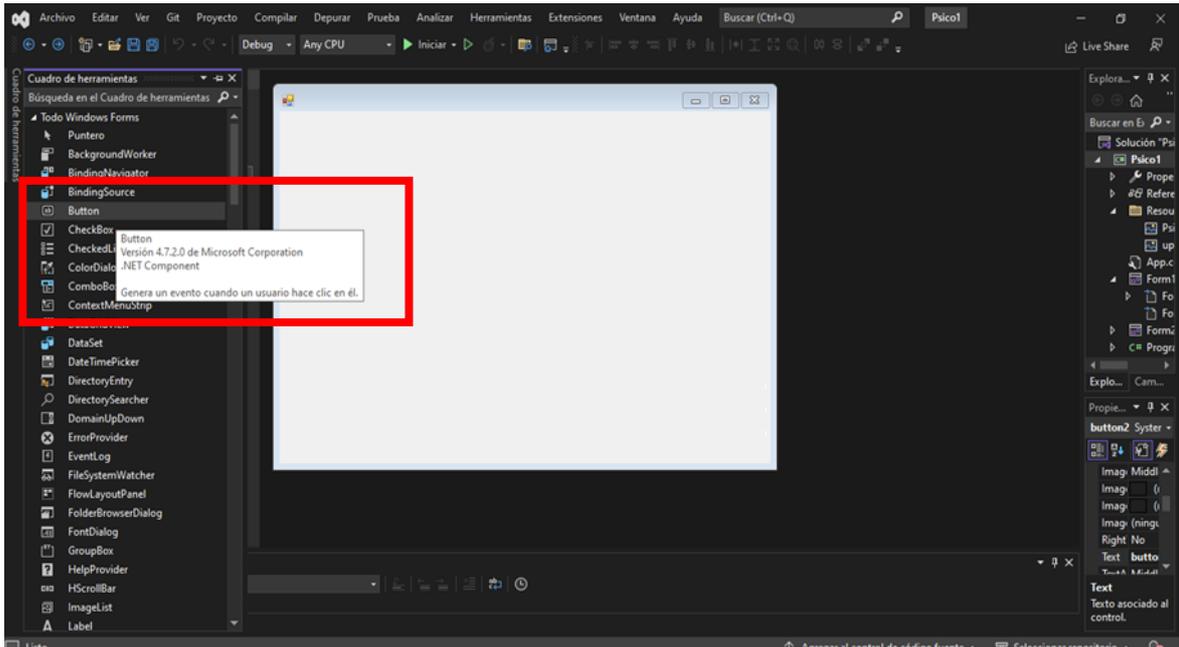
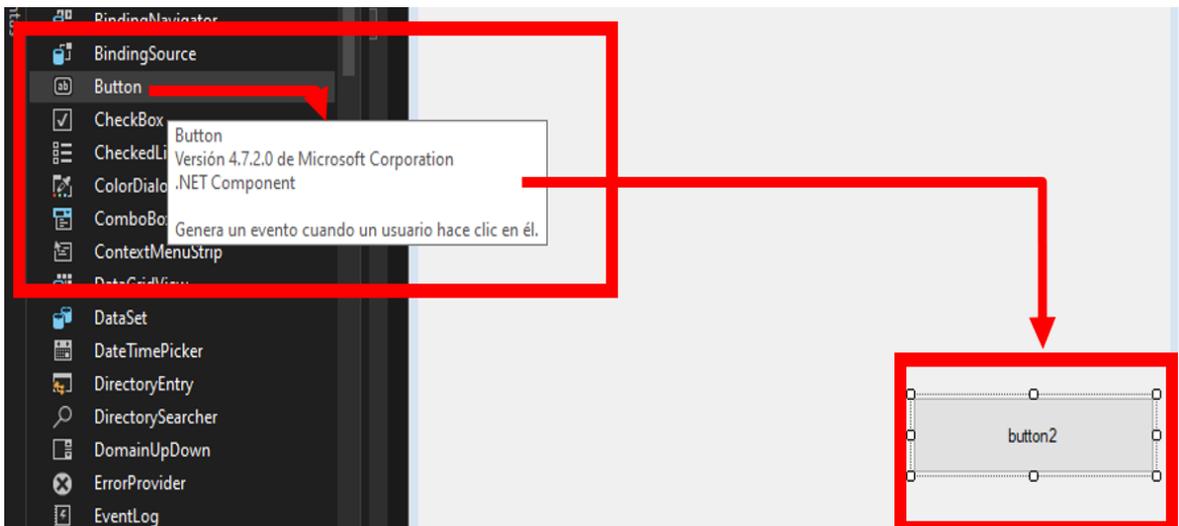


Figura 1.9: Creación del “Button”



- Usando las mismas herramientas que se aplicaron anteriormente en la creación del boceto de la pantalla de inicio, se procede a elaborar la pantalla donde ira el test de reacción simple, se escogió el diseño deseado para la elaboración de dicho boceto y de igual manera se coloca un botón donde permite iniciar y finalizar el test.

Figura 1.10: Pantalla del test de reacción simple.

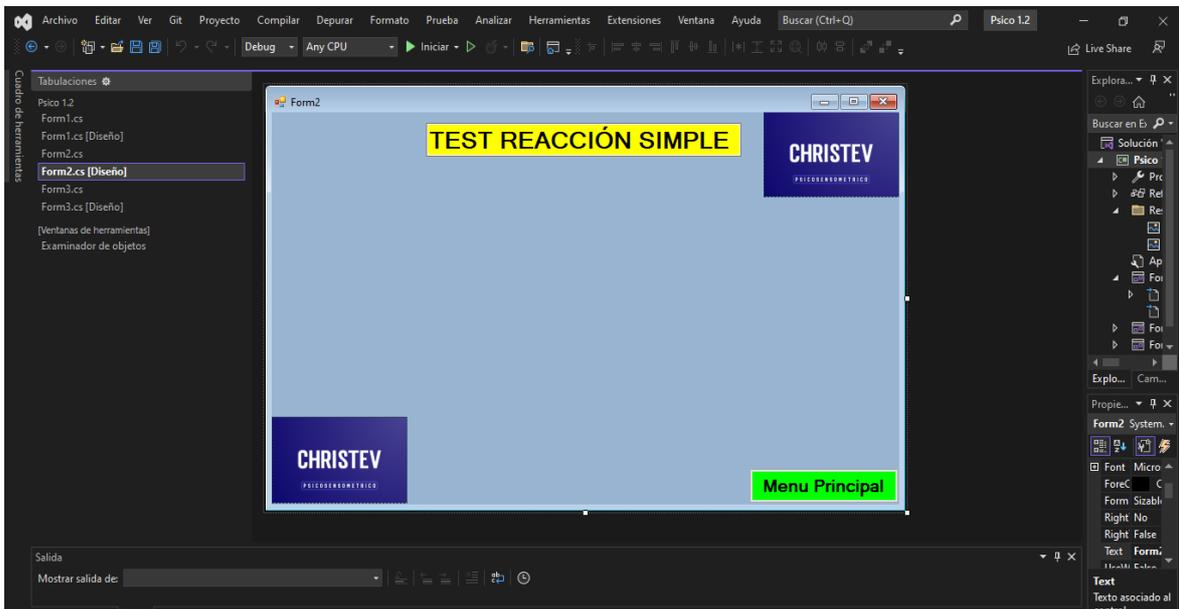
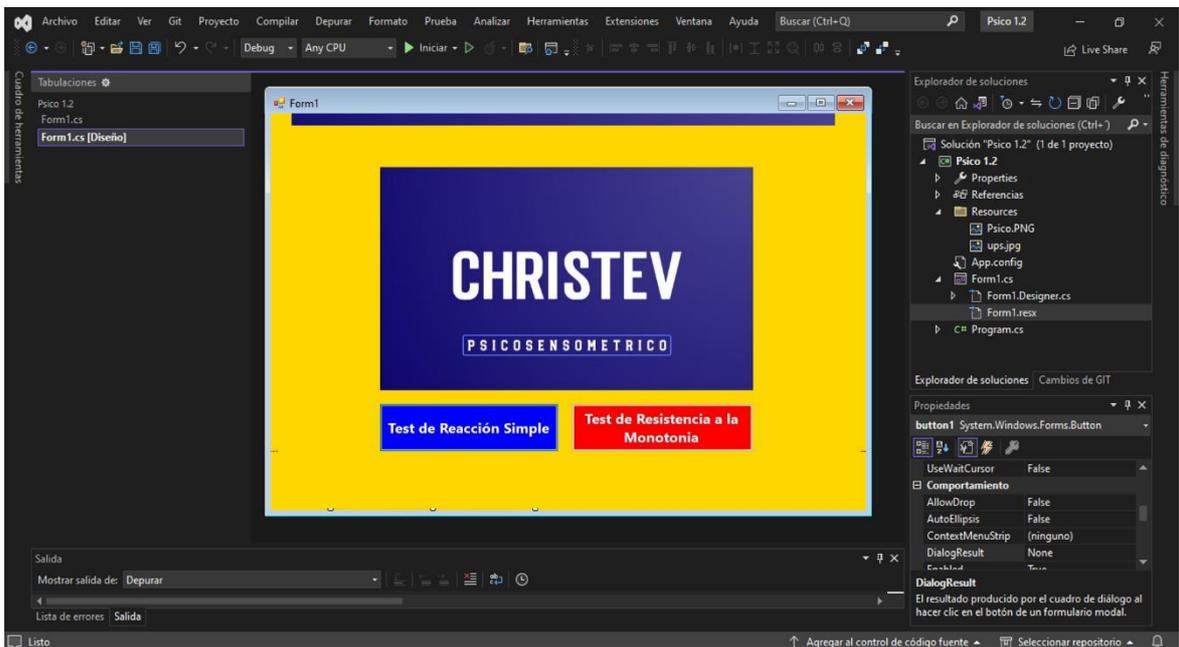


Figura 1.11: Presentación de la pantalla principal con los botones de ingreso a los test



1.2. Hardware

El hardware está representado por la parte física del simulador, ya que esto facilita el funcionamiento del programa en el ordenador, para el presente proyecto se utilizará un volante de juego y sus pedales los cuales se detallan a continuación:

1.2.1. Volante y pedales

El Genius TwinWheel F1, es un volante de carreras 2 en 1 para PS2 y PC, y es justo el volante que se necesita para mantener el control mientras se juega cualquier videojuego de carreras, especialmente el volante seleccionado, cuenta con la tecnología TouchSense patentada de inmersión y retroalimentación de vibración avanzada para experimentar las diferentes sensaciones que genera en cada juego.

Incluye un panel direccional y cuatro botones de control para controles avanzados, como el control del volante o la bocina, y cambiar la pantalla según los requisitos del software del juego. El control del pedal del freno y del acelerador proporciona una experiencia de conducción realista.

Figura 1.12: Volante y Pedal Simulador



1.2.2. TwinWheel F1 es un volante especializado diseñado por Speedlink para juegos de simulación de carreras, centrado especialmente en ofrecer una experiencia realista para simuladores de Fórmula 1. Este dispositivo incluye características como feedback de fuerza para simular condiciones de conducción realistas y pedales, proporcionando a los usuarios un control preciso y una inmersión profunda en la experiencia de conducción virtual.

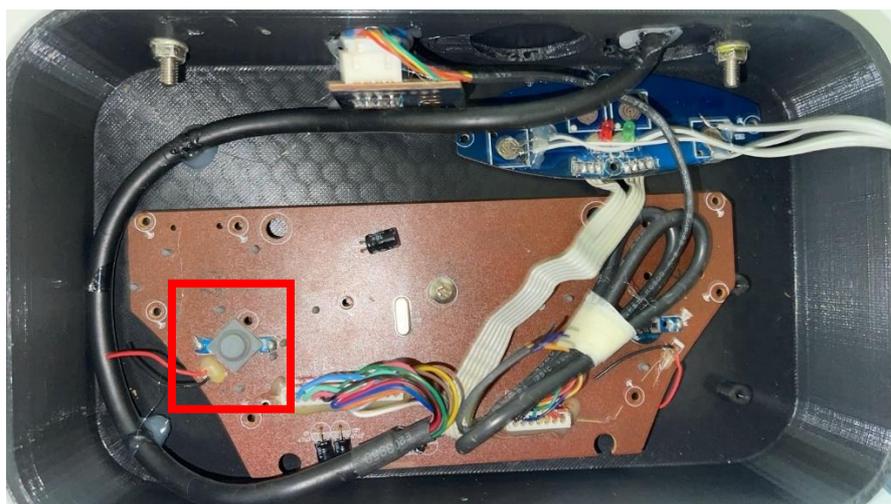
1.2.3. Desarmado del volante

Para asegurar el funcionamiento óptimo de la pedatera TwinWheel F1, es fundamental comenzar con el despiece meticuloso del volante de juegos. Este proceso comienza por la base del volante, un componente clave que proporciona estabilidad y soporte esencial para la manipulación adecuada del mismo. Fabricada con plástico resistente y partes de metal, la

base está diseñada con almohadillas antideslizantes en su parte inferior para evitar movimientos no deseados durante su uso intensivo.

La base del volante también está equipada con abrazaderas ajustables o puntos de montaje, permitiendo su fijación segura a una mesa o cualquier superficie de juego. Posteriormente, se procede al desmontaje de los botones y perillas adicionales del volante de juegos. Estos elementos suelen estar estratégicamente ubicados en el volante mismo, en la base o incluso en la palanca de cambios, contribuyendo así a una experiencia de usuario más completa y personalizable en simuladores de conducción de alta precisión como los juegos de Fórmula 1.

Figura 1.13: Volante desarmado



Estos elementos permiten al jugador acceder a funciones específicas del juego, como activar comandos, cambiar de vista, ajustar la configuración del volante, etc., se retira todas las conexiones innecesarias.

Las partes obtenidas del volante de juegos se conectan a la computadora a través de cables, la conexión más común es mediante un cable USB, aunque también puede haber opciones inalámbricas disponibles. Además, algunas partes del volante pueden requerir una fuente de alimentación adicional para funcionar correctamente.

Figura 1.14: Partes del volante a ser utilizado



La placa cumple un papel crucial al facilitar el control del dispositivo. Conectada a la computadora a través de un cable USB y a la pedalera mediante un cable de audio, asegura el funcionamiento coordinado de ambos componentes. Durante las pruebas, la pedalera será utilizada para evaluar la reacción simple, mientras que la placa será sometida al test de resistencia a la monotonía. Para optimizar la estética del simulador, la placa será alojada en una caja diseñada específicamente con este propósito.

Figura 1.15: Pulsadores del Simulador



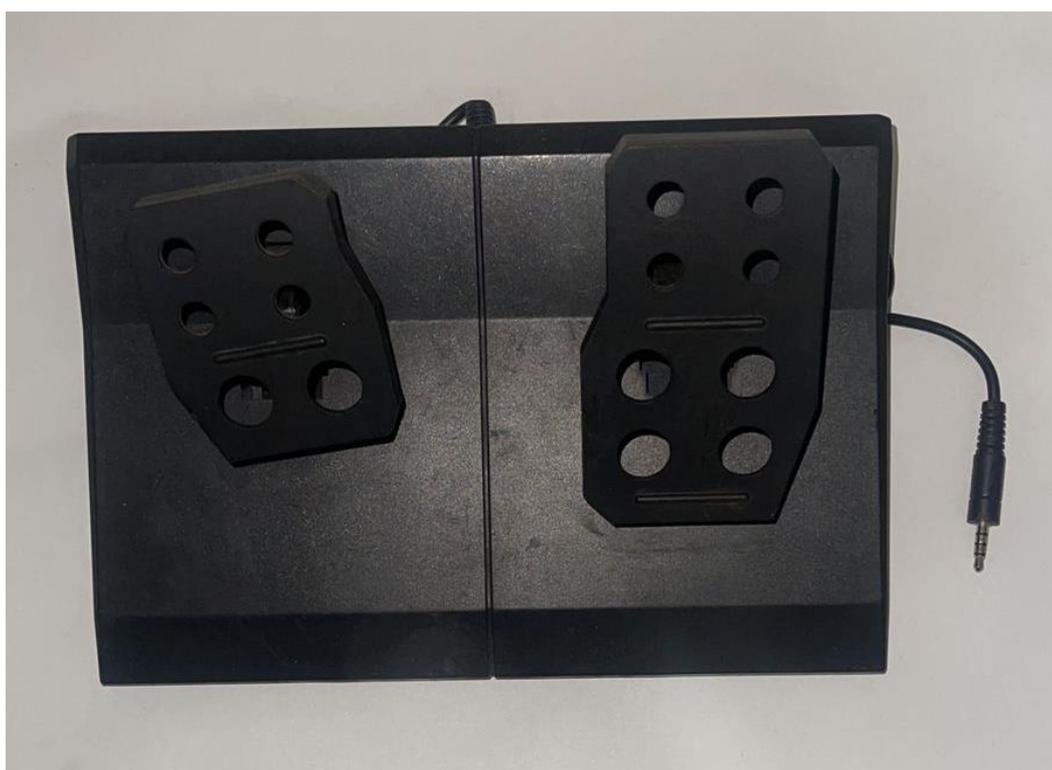
1.2.4. Pedales: Mecanismo

Los pedales son elementos utilizados en las pruebas para evaluar la habilidad y capacidad de los individuos en tareas que requieren coordinación y control, así como para medir su

tiempo de reacción. Estas pruebas son comúnmente utilizadas en diversos ámbitos, como la psicología, la medicina laboral y el ámbito de la seguridad vial.

Se utilizarán estos pedales para el proyecto y así poder manipular mejor los test, además de conocer el nivel de rendimiento que tiene cada uno de los usuarios del simulador de conducción, esto es importante para conocer sus actitudes psico-sensoriales y el tiempo de reacción de cada uno en la pulsación del pedal con el pie derecho ya que se utiliza el freno y el acelerador esto ayuda a ver el tiempo de reacción que se tiene al momento de conducir y en caso de presentarse un semáforo, algún peatón imprudente o un disco de pare se identifique inmediatamente y se establecerá el tiempo de reacción en segundos los cuales por medio de 10 intentos se realizara un promedio y se conocerá si pasa o no la prueba.

Figura 1.16: Pedales Simulador.



Las placas electrónicas están compuestas principalmente por un sustrato aislante, generalmente de fibra plástica para poder programar la pedalera, ya que se conecta directamente a la placa para su funcionamiento y así poder trabajar adecuadamente con los pedales y el programa de simulación de conducción.

1.2.5. Interfaz: Conexión

Con ayuda de un cable HDMI transmitimos señales de audio y video digital de alta calidad que se verán reflejados en el monitor.

Figura 1.17: Cable de audio que va conectado desde los pedales a la placa



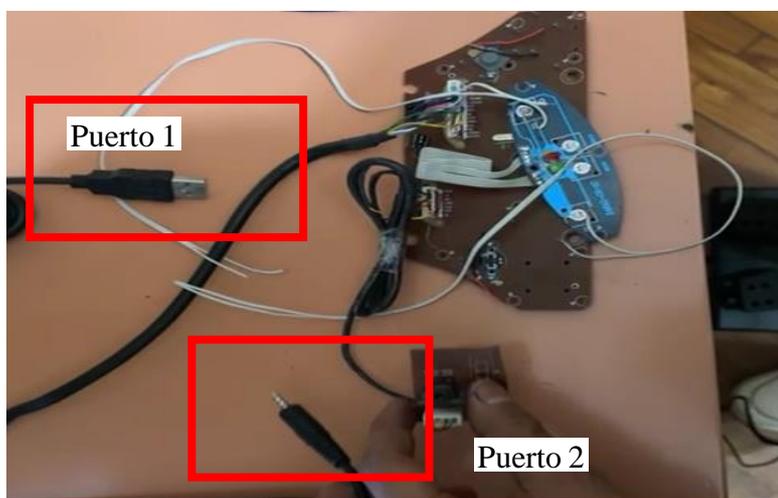
Nos valemos de cables USB 3.0 para la transferencia de datos y la conexión de todos los dispositivos electrónicos de la pedalera.

Figura 1.18: Cable USB que se conecta al computador



Los cables que van conectados por medio del USB (puerto 1) a la computadora y el cable de audio que va conectado a la pedalera (puerto 2), para su funcionamiento.

Figura 1.19: Conexión de los cables



1.3. Test de prueba de reacción simple.

El test de prueba de reacción simple controla la capacidad de responder a estímulos que implican respuestas simples, sencillas y automáticas. En psicología, el tiempo de reacción se define como “el tiempo entre la aparición de un estímulo sensorial y una respuesta

voluntaria de los músculos”, lo que se observa es la respuesta juzgadora del sujeto, por ejemplo, cuando percibe el estímulo decide si o no pisar el pedal, por lo que en la situación de estudio se observan acciones precipitadas tales como pisar/soltar prematuramente el pedal antes de que aparezca el estímulo (Biartic, 2022a).

Medir estas reacciones es importante para la evaluación de los conductores, ya que nos permiten distinguir a los conductores cuyos tiempos de reacción son demasiado lentos de aquellos que reaccionan impulsivamente.

El objetivo es evaluar la velocidad de reacción que tiene la persona ante un estímulo simple. Se le debe indicar a la persona que se utilizará la pedalera para realizar el examen y procedemos a hacer clic sobre el botón "Realizar examen".

Esto se logra mediante un examen en donde se muestra el habitáculo de un vehículo inicialmente detenido.

Le pedimos al examinado que presione el pedal derecho (tal cual se lo indica en la interfaz de la pantalla) y lo mantenga presionado, en este momento el vehículo entrara en marcha.

Si la persona suelta el pedal derecho antes de tiempo, se le mostrara un aviso debajo informándole que no debe soltarlo.

Entonces, la persona presiona el pedal derecho, y espera hasta que la señal de PARE aparezca, en este momento debe soltar el pedal derecho y presionar el izquierdo lo más rápido posible, este test debe realizarse solo con el pie derecho.

Una vez que se procede a realizar el test, el tiempo de respuesta se muestra en la pantalla del examinador y se presiona el pedal derecho para continuar la prueba, esto significa que se acelera y cuando aparece un rotulo de pare se debe frenar inmediatamente. Continuando hasta completar 10 repeticiones del test. El número de repeticiones restantes se muestra al finalizar el test, los resultados se mostrarán en la pantalla de vista previa con la selección de resultados habilitada, el campo de monitoreo y el botón Guardar examen. Después de ingresar la información relevante, se continúa haciendo clic en el botón "Guardar".

En la obtención de los resultados el valor normal de la velocidad de respuesta promedio debe ser inferior a 0,435 segundos; de lo contrario, el resultado rápido "fallará".

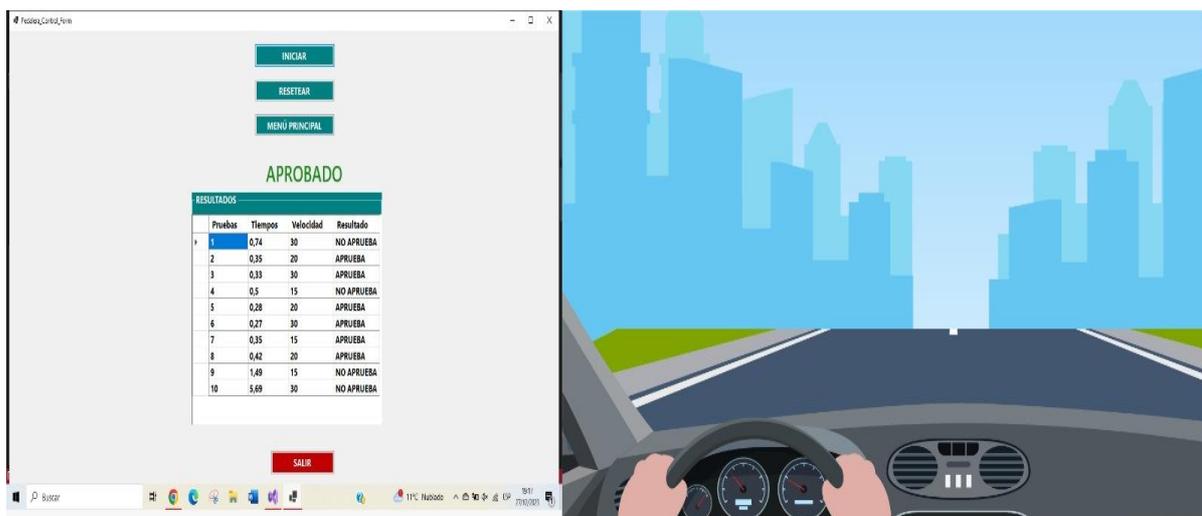
Para la realización del test de prueba de reacción simple se evalúa la reacción del usuario con el pie derecho al momento de utilizar el acelerador y el freno. (Reglamento General de la Ley de Tránsito y Transporte Terrestre del 28 de enero de 1997)

Figura 1.20: Pedales y maniobras con el pie derecho



Con la realización del test se obtendrán 10 valores diferentes los cuales permitirán obtener un promedio, permitiendo verificar que el valor de tiempo que se toma en acelerar y parar al momento que el sistema lo solicita, para así reunir una serie de pruebas sacando un promedio y verificar si aprueba o no, el valor máximo que puede sacar es 0,43 segundos, y si el promedio de los 10 intentos da un valor menor al valor máximo aprueba, pero si saca mayor al valor establecido en segundos reprobara y tendrá que seguir practicando o a su vez someterse a un examen médico, para determinar el motivo de su reacción retardada ante los eventos. (Ley de Tránsito y Transporte Terrestre del 2 de agosto de 1996)

Figura 1.21: Resultados del test (promedio)



1.4. Programación del simulador de conducción

Todos los programas fueron creados utilizando el lenguaje de programación C# en el entorno de desarrollo Visual Studio. Dentro de Visual Studio, se utilizó la plataforma Windows Forms para diseñar las interfaces gráficas de usuario (GUI) de los programas. Windows Forms proporciona un conjunto de clases y controles que permiten a los desarrolladores crear interfaces de usuario intuitivas y dinámicas para aplicaciones de escritorio. Estas interfaces de usuario se componen de ventanas rectangulares, llamadas 'formularios', en las cuales se pueden agregar y configurar diferentes elementos gráficos y objetos interactivos, como botones, cajas de texto, cuadros combinados y muchos otros controles. Los formularios pueden responder a eventos del usuario, como clics de ratón y pulsaciones de teclas, y pueden interactuar con la lógica del programa, que se implementa utilizando el lenguaje de programación C#. En resumen, el desarrollo de los programas implica la combinación del poder de C# como lenguaje de programación, la versatilidad de Windows Forms para la creación de interfaces de usuario y las herramientas integradas de Visual Studio para facilitar el proceso de desarrollo. (Microsoft, 2022).

Con Windows Forms se puede desarrollar aplicaciones con gráficos que sean fáciles de implementar y actualizar, y que se puedan usar sin conexión a internet. Las aplicaciones de Windows Forms pueden acceder al hardware local y al sistema de archivos de la computadora en la que se ejecutan.

Según Besoftware (2022), C# (pronunciado como C Sharp) es un lenguaje de programación desarrollado por Microsoft que fusiona lo mejor de los lenguajes C y C++, incorporando

constantemente nuevas características y adoptando parte de la sintaxis avanzada de otros lenguajes como Java. Este lenguaje introdujo la programación orientada a objetos en toda la plataforma (tanto en su Framework como en su versión Core) y ha ido adaptando herramientas de generación de código de otros lenguajes populares, como Visual Basic. Esto ha hecho de C# un lenguaje versátil y fácil de aprender, manteniendo la robustez del C original. Además, su compilador ha sido completamente reconstruido, lo que ha llevado a mejoras significativas en el rendimiento de las aplicaciones, con un aumento de hasta un 600% en la velocidad de ejecución.

Figura 1.22: Programación C#



```
C# Copiar
using System;

class Hello
{
    static void Main()
    {
        Console.WriteLine("Hello, World");
    }
}
```

Usamos la directiva 'using' sirve para importar librerías externas dentro de un programa, lo que permite acceder a sus funcionalidades en el código. Cada vez que se emplea 'using', se está incorporando una librería específica al entorno de trabajo del programa. Por otro lado, un 'namespace' (espacio de nombres) en C# es un contenedor que organiza y agrupa un conjunto de elementos relacionados, como clases, interfaces y otros tipos. Sirve para evitar conflictos de nombres y proporciona un contexto claro para las diferentes partes del código. En resumen, mientras que 'using' se encarga de incluir funcionalidades externas, los 'namespaces' estructuran y organizan el espacio de trabajo del programa.

Los lenguajes de programación más antiguos como C, Basic o COBOL tienen código de procedimiento; es decir, se programa y ejecuta paso a paso una serie de instrucciones secuenciales. Incluso si contienen subrutinas o funciones, es difícil aislar datos específicos porque toda gira en torno a la lógica. La razón principal detrás de la elección de la programación orientada a objetos (POO) radica en su capacidad para organizar el diseño de software alrededor de datos y objetos, en lugar de centrarse únicamente en funciones y lógica. Según IBM (2021), POO se caracteriza por su enfoque en aquello con lo que el

programador necesita trabajar, permitiendo definir objetos que poseen propiedades y comportamientos únicos. Estos objetos se estructuran en tipos o clases que pueden heredar propiedades de categorías superiores o super-categorías, lo que facilita la organización y reutilización del código. Esta metodología es especialmente útil en programas de gran escala y complejidad, ya que permite una gestión más eficiente y una estructura más modular, lo que facilita las actualizaciones periódicas del software.

Se usa la declaración de variables globales para almacenar datos accesibles en todo el programa, es fundamental comprender los tipos de variables disponibles en C#. Entre los tipos más comunes se encuentran 'int' para enteros, 'double' para números de punto flotante y 'string' para cadenas de caracteres. Además, para almacenar conjuntos de datos más complejos, se recurre a los arreglos, representados por corchetes. De esta manera, se puede declarar un arreglo especificando el tipo de datos que contendrá, seguido de corchetes para indicar que se trata de un arreglo. Por ejemplo, para declarar un arreglo de tipo entero en C#, se utilizaría la siguiente sintaxis: 'int [nombreDelArreglo;]'. Esta declaración prepara el espacio de memoria necesario para almacenar una secuencia de números enteros, que luego pueden ser accedidos y manipulados en diferentes partes del programa.

Luego de la declaración de variables globales, procedemos a definir variables booleanas, las cuales pueden tener únicamente dos valores: verdadero o falso. Posteriormente, utilizamos el tipo de dato 'String' para declarar variables destinadas a almacenar texto. Además, empleamos la biblioteca 'Bitmap', la cual nos permite proyectar imágenes dentro de los formularios. En esta etapa, se declaran las imágenes específicas que se utilizarán en el formulario en cuestión. De esta manera, mediante la definición de variables booleanas para representar estados binarios y el uso de 'String' para manejar texto, así como la inclusión de imágenes a través de la biblioteca 'Bitmap', logramos preparar el entorno necesario para el funcionamiento del programa.

En la sección de declaración de imágenes, se aborda todo lo relacionado con el manejo de imágenes utilizando la biblioteca Bitmap. Aquí se realiza la llamada específica a cada imagen que será utilizada en el formulario. Además, se establece el modificador de acceso 'Public', lo que significa que todas las formas (forms) pueden acceder a esta función. Se utiliza 'KeyPress' para mapear la entrada del teclado y convertirla en una función que

permite al programa detectar los pulsos de la pedalera en los botones correspondientes. Se define 'Bounds' utilizando 'Screen.PrimaryScreen.Bounds', lo que permite que la aplicación se ajuste automáticamente al tamaño y la ubicación de la pantalla principal. Asimismo, se asigna la ubicación de proyección del formulario actual mediante la propiedad 'Screen'. De esta manera, se establecen las configuraciones necesarias para el correcto funcionamiento del programa, garantizando la correcta visualización de las imágenes y la detección precisa de las entradas del teclado.

Programación paso a paso.

La programación de la pedalera en el simulador de conducción constituye un elemento central en la implementación práctica de esta herramienta educativa. A través del uso de Visual Studio y el lenguaje de programación C#, se ha desarrollado una interfaz interactiva que simula de manera precisa diversas situaciones de conducción. Este componente esencial del simulador permite la interacción del usuario con el entorno virtual. En este apartado, se detalla el proceso de programación detrás de la pedalera, incluyendo la manipulación de eventos, el dibujo de elementos visuales y la gestión de variables para garantizar un funcionamiento óptimo y realista del simulador.

private void Pedalera_Test_Paint (object sender, PaintEventArgs e): En esta línea de programación se ejecuta un método encargado de dibujar y proyectar todas las imágenes necesarias según la lógica del programa, asociado al evento 'Paint', se activa cada vez que se necesita actualizar o redibujar el contenido visual del formulario 'Pedalera_Test'. En este proceso, se utiliza el objeto 'e' de tipo 'PaintEventArgs' para acceder al lienzo de dibujo y realizar las operaciones necesarias para mostrar gráficamente los elementos definidos en el programa. De esta manera, se asegura que la interfaz del usuario refleje correctamente el estado actual de la aplicación, mostrando las imágenes y elementos visuales según lo establecido en el programa.

'carro = Properties.Resources.carro;' en esta línea se realiza una acción crucial al referenciar imágenes almacenadas en la carpeta de recursos ('Resources') del proyecto. Esta práctica asigna la imagen designada como 'carro' a la variable 'carro'. Aprovechando la propiedad 'Resources' provista por el entorno de Visual Studio, accedemos de manera directa a las imágenes y otros recursos previamente integrados en el proyecto. De esta manera nos permite emplear las imágenes de manera eficiente dentro del código, evitando la necesidad

de manipular rutas de archivo externas. Con ello, se logra una gestión más organizada y simplificada de los recursos visuales empleados en nuestra aplicación, garantizando una experiencia de desarrollo más fluida y eficaz.

Graphics.DrawImage (carretera, 0, 0); esta línea de código se encarga de dibujar una imagen en el lienzo de dibujo utilizando un objeto de tipo gráfico. En este caso, se dibuja la imagen denominada 'carretera' en la posición (0, 0) del lienzo. La función DrawImage() toma como argumento la imagen que se va a dibujar ('carretera') y las coordenadas (x, y) donde se ubicará la esquina superior izquierda de la imagen en el lienzo.

private void Pedalera_Test_Load (object sender, EventArgs e)

Inicializar(); :La función 'Pedalera_Test_Load' se activa cuando el formulario 'Pedalera_Test' se carga por completo. En esta línea de código, se llama a la función 'Inicializar()' para realizar las configuraciones iniciales necesarias al cargar el formulario. Esto garantiza que el formulario esté listo para su uso y que todas las operaciones previas a la interacción del usuario se completen de manera adecuada.

public void Reseteo_Control() La función 'Reseteo_Control()' se encarga de reiniciar el control a su estado inicial. Dentro de esta función.

contador_texto = 0; 'contador_texto' se establece en 0, lo que indica que se reinicia el contador de texto utilizado en el control.

ReseteoTemporizador(); Se llama a la función 'ReseteoTemporizador()', que se encarga de reiniciar cualquier temporizador utilizado en el control.

contador = 0; 'contador' se reinicia a 0, lo que indica que se restablece cualquier otro contador utilizado en el control.

resultados.Clear(); El contenedor llamado 'resultados' se limpia, eliminando cualquier dato almacenado previamente.

bandera_habilita = false; 'bandera_habilita' se establece en false, lo que indica que se deshabilita al reiniciar el control.

En resumen, esta función asegura que todas las variables y elementos del control estén listos y en su estado inicial para un nuevo programa o para su reinicio completo.

private void button1_Click(object sender, EventArgs e) En esta línea de código la función 'button1_Click' es un controlador de eventos que se activa cuando se hace clic en el botón asociado en la interfaz de usuario. Dentro de esta función.

IniciarTemporizador(); 'IniciarTemporizador()' se llama para iniciar cualquier temporizador o contador que pueda ser necesario para el examen o proceso que se va a realizar. Esto puede ser útil para establecer límites de tiempo o para realizar ciertas acciones en intervalos regulares.

IniciarExamen = true; 'IniciarExamen' se establece en true, lo que indica que se está iniciando el examen o proceso importante dentro del programa. Esta variable puede ser utilizada para controlar el flujo del programa y para habilitar ciertas funcionalidades específicas relacionadas con el inicio del examen.

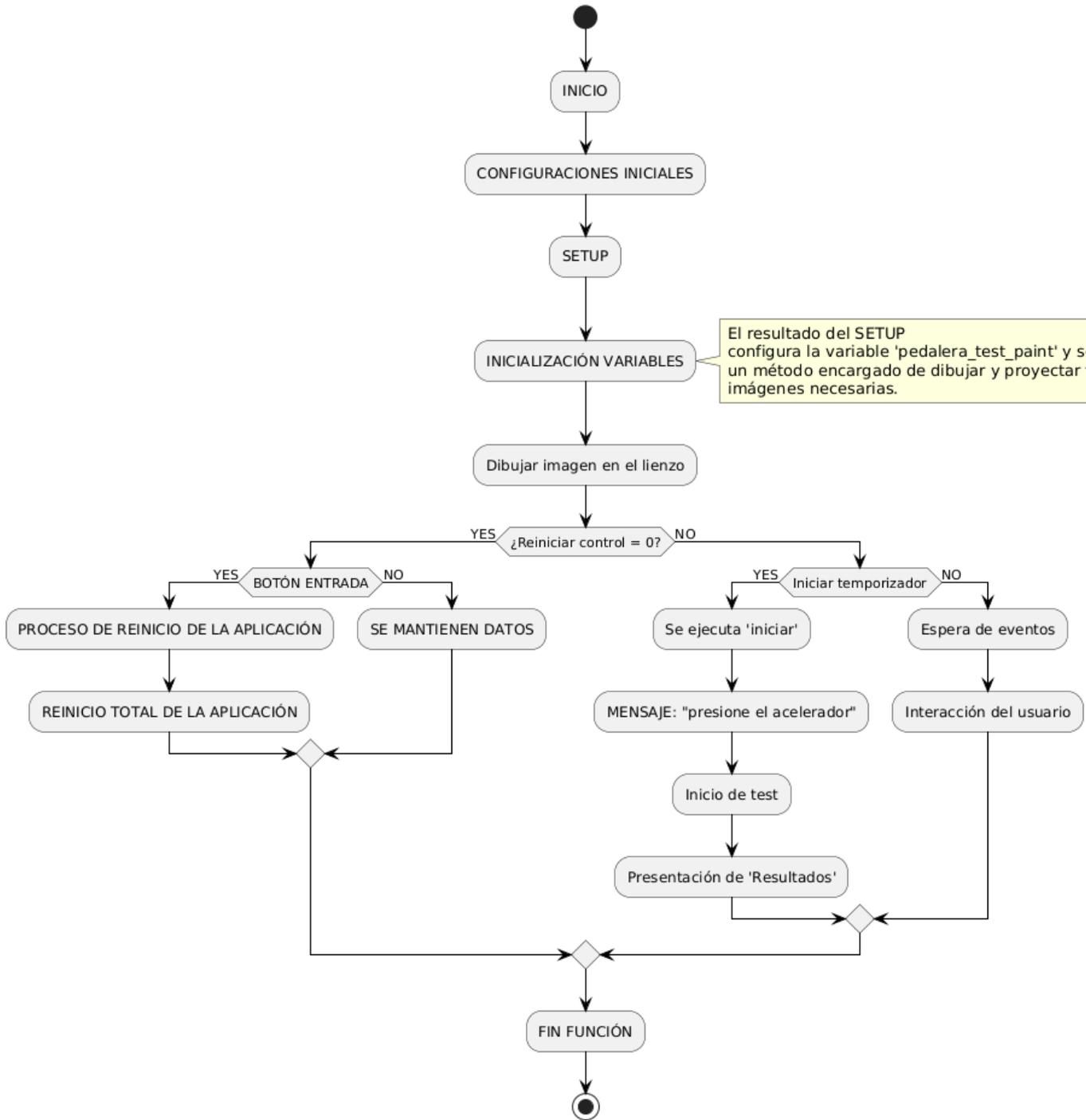
Además, dentro de esta función, se pueden incluir otras funciones necesarias para preparar el entorno o realizar alguna acción específica al iniciar el examen.

En resumen, esta función encapsula todas las acciones necesarias cuando se hace clic en el botón de inicio, lo que incluye la inicialización de temporizadores y la configuración de variables para iniciar un examen o proceso dentro del programa.

public void iniciar() Dentro de la función 'iniciar()', se realiza lo siguiente:

label2.Visible = true; 'label2.Visible = true;' establece la propiedad 'Visible' del control 'label2' en true, lo que significa que se hace visible en la interfaz de usuario. Esto es útil para mostrar información o elementos visuales que inicialmente estaban ocultos pero que deben ser visibles al iniciar el programa.

PROGRAMACION DEL SIMULADOR DE CONDUCCION 'TEST DE REACCION SIMPLE'



//IniciarTemporizador();

IniciarExamen = true; 'IniciarExamen = true;': asigna el valor true a la variable booleana 'IniciarExamen'. Esto indica que se está iniciando el examen dentro del programa. Esta variable es utilizada para controlar el programa y para habilitar funciones específicas relacionadas con el inicio del examen.

En resumen, esta función 'iniciar()' se encarga de realizar acciones al iniciar el programa, como hacer visible un elemento en la interfaz de usuario y establecer una variable booleana para indicar que se está iniciando un examen o proceso.

private void FrenarVehiculo() Dentro de la función 'FrenarVehiculo()', se realiza lo siguiente:

if (IniciarExamen && Detenido == true) La condición 'if (IniciarExamen && Detenido == true)' verifica dos condiciones dentro del bloque del código if:

Esta variable booleana indica que se ha iniciado el examen o proceso dentro del programa. Y con la variable "Detenido" indica si el vehículo está detenido. Si ambas condiciones son verdaderas, significa que se ha iniciado el examen y el vehículo está detenido, se ejecuta el bloque de código dentro del 'if'.

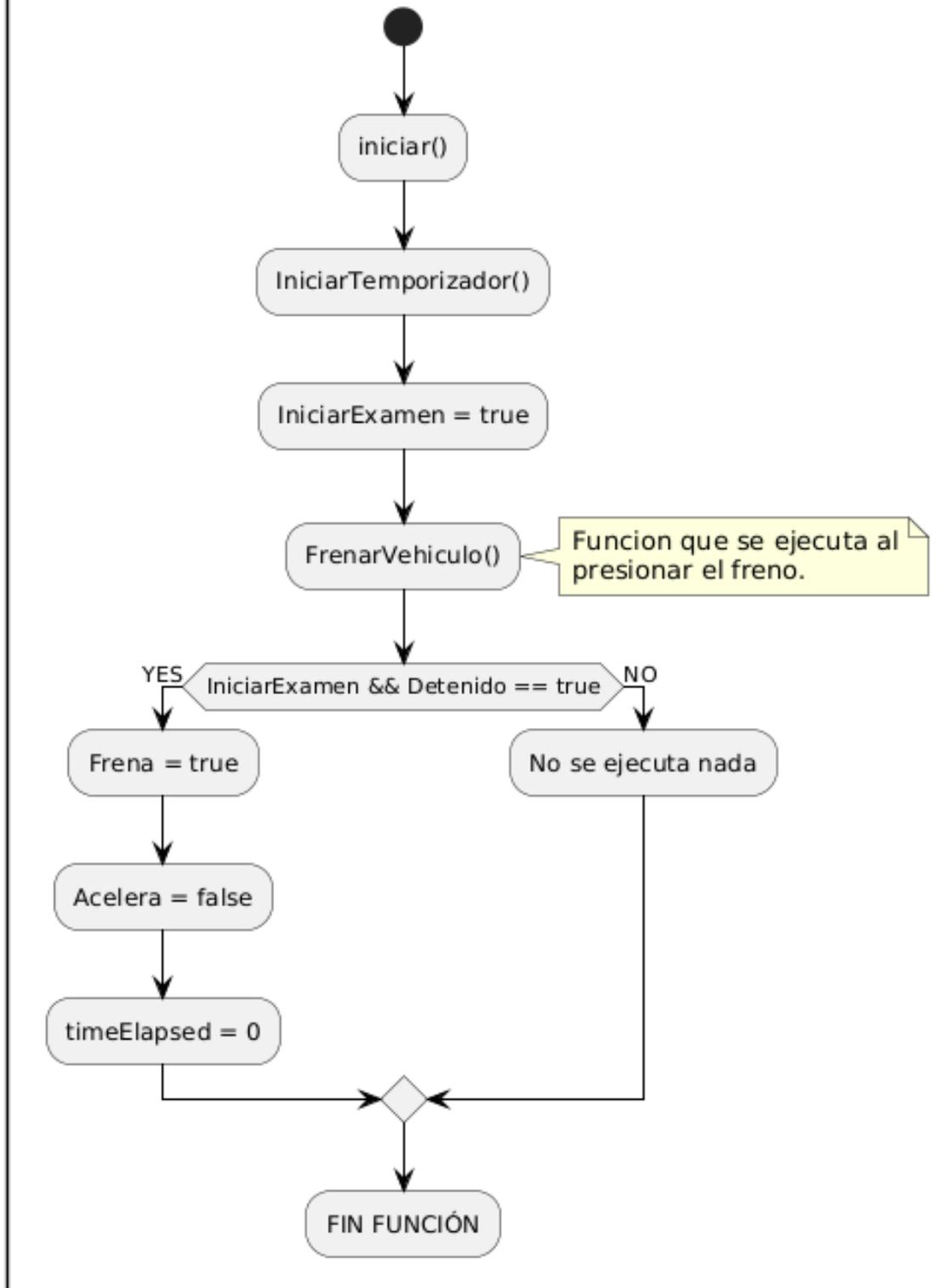
Frena = true; 'Frena = true;': Se establece la variable booleana 'Frena' en true, lo que indica que el vehículo frena.

Acelera = false; 'Acelera = false;': Se establece la variable booleana 'Acelera' en false, lo que indica que el vehículo no está en movimiento.

timeElapsed = 0; Se reinicia el contador de tiempo 'timeElapsed' a 0, lo que indica que se está comenzando a contar el tiempo transcurrido desde que el vehículo freno.

En resumen, esta función 'FrenarVehiculo()' se encarga de ejecutar las respectivas acciones cuando se presiona el freno del vehículo, como indicar que el vehículo está detenido, pausar la aceleración y reiniciar el contador de tiempo en ciertas condiciones específicas dentro del programa.

Programación de Inicio y Frenado del Vehículo



public void IniciarTemporizador() Dentro de la función 'IniciarTemporizador()', se realiza lo siguiente:

SwTiempoReaccion.Start(); Se inicia el temporizador 'SwTiempoReaccion'. Este temporizador se utiliza para medir el tiempo de reacción del usuario desde que aparece la señal de "pare" hasta que el usuario presiona el freno. Al iniciar este temporizador, se comienza a contar el tiempo transcurrido.

timer1.Enabled = true; Se activa el temporizador 'timer1'. Este temporizador se utiliza para controlar el movimiento del camino. Al activar este temporizador, se inicia su conteo y se ejecutan las acciones asociadas a él en intervalos regulares.

En resumen, esta función 'IniciarTemporizador()' se encarga de iniciar los temporizadores necesarios para medir el tiempo de reacción del usuario, controlar el movimiento del camino y realizar otras acciones relacionadas con el tiempo dentro del programa cuando se activa el freno del vehículo.

private void RealizarPrueba(); Esta función contiene toda la lógica principal del programa. Dentro de esta función, se utiliza una estructura condicional 'if' para realizar ciertas acciones dependiendo de una condición específica. En este caso:

Se utiliza un condicional 'if' para verificar si la variable 'contador' es igual a 10. Esto significa que se ha alcanzado cierto punto o etapa importante en la ejecución del programa.

Si la condición se cumple (es decir, si 'contador' es igual a 10), se ejecuta el bloque de código dentro de la llave '{}'.
'{ }'

Dentro de este bloque de código, se realizan ciertas acciones relacionadas con el avance del programa o el procesamiento de datos.

Además, en esta parte del código, se incluye la pausa de un temporizador u otras operaciones necesarias para controlar el flujo del programa.

Por otro lado, si la condición del 'if' no se cumple (es decir, 'contador' no es igual a 10), se puede definir una lógica alternativa utilizando un bloque 'else'. Esto se realizaría para manejar cualquier otro escenario que no cumpla la condición establecida en el 'if'.

Por ejemplo, la frase "SI bandera no aplasta y bandera_habilita" es interpretada como una condición adicional que se evalúa junto con el contador. Si estas banderas no están activadas o no cumplen ciertas condiciones, entonces se ejecutaría la lógica dentro del 'else' o del bloque alternativo.

En resumen, esta línea de código dentro de la función 'RealizarPrueba()' establece la lógica principal del programa basada en la evaluación de una o más condiciones específicas. Dependiendo del resultado de estas condiciones, se ejecutarán ciertas acciones definidas dentro de la función.

if (bandera_noaplasta && bandera_habilita); bandera_noaplasta: Es una variable booleana que indica si el freno no está siendo presionado. En cambio, "bandera_habilita" es una variable booleana que indica si ciertas condiciones para habilitar el movimiento están cumplidas.

En resumen, esta condición verifica que ambos valores booleanos sean true para ejecutar el bloque de código dentro del if. Es decir, el programa solo ejecuta el código siguiente si el freno no está presionado y si otras condiciones habilitantes están cumplidas.

tiempo_segundo = tiempo_segundo + velocidad; tiempo_segundo es una variable que representa el tiempo o una medida del progreso del camino en tu simulador. Por otra parte, velocidad es una variable que representa la rapidez a la que se mueve el camino.

En resumen, se incrementa el valor de "tiempo_segundo" en una cantidad igual a velocidad. Este incremento simula el movimiento del camino en el tiempo.

if (tiempo_segundo > 140) { tiempo_segundo = -100; Verifica si "tiempo_segundo" ha excedido el valor de 140. Este cambio representa que el camino se ha desplazado completamente y debe comenzar desde una posición inicial.

//timeElapsed += 50; Variable que representa el tiempo transcurrido.

label1.Text = timeElapsed.ToString(); Convierte el valor de timeElapsed a una cadena de texto.

Invalidate(); Marca el control como inválido, lo que provoca que se vuelva a repintar en la siguiente actualización del sistema de ventanas. Esto es útil para actualizar la visualización del control después de cambiar sus propiedades o datos mostrados.

En pocas palabras esta línea de código está en un método que se ejecuta periódicamente (un temporizador). Controla la lógica del movimiento del camino en el simulador de conducción, actualizando tanto la posición del camino como la interfaz gráfica para reflejar el tiempo transcurrido. El uso de las banderas booleanas permite condicionar la ejecución del código a ciertos estados del simulador (como si el freno no está siendo presionado)

if (contador <= TiemposPrueba.Length && ts.Seconds >= TiemposPrueba[contador])

contador: Es una variable que se utiliza para llevar un registro de la posición actual en la serie de tiempos de la prueba.

TiemposPrueba: Es un arreglo que contiene tiempos específicos en los que se debe realizar alguna acción (en este caso, visualizar la imagen de "PARE").

ts.Seconds: es una propiedad del objeto ts (tipo TimeSpan), que representa la cantidad de segundos transcurridos desde el inicio o desde el último reinicio del contador.

Condicional if explicado:

contador <= TiemposPrueba.Length: Verifica que el valor de contador no exceda el tamaño del arreglo TiemposPrueba, asegurando que aún hay elementos válidos en el arreglo para comparar.

ts.Seconds >= TiemposPrueba[contador]: Compara los segundos transcurridos (ts.Seconds) con el tiempo específico almacenado en TiemposPrueba en la posición indicada por contador. Si los segundos transcurridos son mayores o iguales al tiempo de prueba en esa posición, la condición se cumple.

PararRecorrido(); Es una función (o método) definida en el programa. Su propósito es detener el recorrido o el movimiento actual en el simulador. Cuando se llama a esta función, el programa ejecutará las instrucciones necesarias para detener el recorrido.

Detenido = true; Establece la variable Detenido a true, indicando que el recorrido ha sido detenido.

En resumen, esta línea de código se ejecuta periódicamente para verificar si ha llegado el momento de realizar una acción específica (en este caso, visualizar la imagen de "PARE"). La condición se basa en el tiempo transcurrido (ts.Seconds) y en una lista de tiempos predefinidos (TiemposPrueba). Cuando el tiempo transcurrido alcanza o supera uno de los tiempos, se llama a la función "PararRecorrido" para detener el simulador y se marca la variable Detenido como true.

Figura 1.23: Programación tiempos de reacción

```

tiempo_segundo = tiempo_segundo + velocidad; // se mueve el camino
if (tiempo_segundo > 140) { tiempo_segundo = -100; } // empieza a moverse el camino
//timeElapsed += 50;
label1.Text = timeElapsed.ToString(); // para aparecer tiempo
Invalidate(); // invalidar el temporizador cuando ya se acabe de cumplir la condicion
}
if (contador <= TiemposPrueba.Length && ts.Seconds >= TiemposPrueba[contador]) //VALIDACION DEL TIEMPO PARA VISUALIZAR LA IMAGE
{
    PararRecorrido();
    Detenido = true;
}
if (contador <= TiemposPrueba.Length && Frena == true && Detenido == true) //CAPTURA EL TIEMPO DE REACCION AL FRENAR EL VEHICUL
{
    pbxMuralla.Visible = false;
    TiemposReaccion[contador] = Double.Parse(TiempoTranscurrido) - TiemposPrueba[contador]; //GUARDA EL TIEMPO DE REACCION CUAN
    Frena = false;
    Detenido = false;
    VerificarMovimiento = false;
    IniciarRecorrido();
    double tiempos_redondeado = Math.Round(Math.Round(TiemposReaccion[contador], 2) - tiempoControl, 2);
    resultados.Add(new Tests() { Pruebas = contador + 1, Tiempos = tiempos_redondeado, Velocidad = velocidad, Result = tiempos
    Pedalera_Control_Form.instance.presenta_resultado();

    PausarTemporizador();
    ReseteoTemporizador();
    contador++;
}

```

- El tiempo de reacción se encuentra en milisegundos. Cada uno de estos se encuentran programados tal y como se puede visualizar en la figura anterior.
- Esto permite establecer el reseteo del freno cada que sea necesario.
- Iniciar recorrido es una función que habilita las líneas del camino cuando se aplica el acelerador.
- Al momento en que finaliza el test se procede automáticamente a la presentación de los resultados, los cuales son presentados al participante y el programa automáticamente da a conocer si aprueba o no.
- El tiempo de redondeo sale en dos decimales dependiendo el procesador de cada PC Y se añaden los resultados en una lista.

if (contador <= TiemposPrueba.Length && Frena == true && Detenido == true);

contador <= TiemposPrueba.Length: Verifica que el valor de contador no exceda la longitud del arreglo "TiemposPrueba". Esto asegura que se está dentro del rango válido del arreglo.

Frena = true: Verifica que la variable Frena sea true, lo que indica que el freno ha sido presionado.

Detenido = true: Verifica que la variable Detenido sea true, indicando que el vehículo está detenido.

pbxMuralla.Visible = false; en esta línea de código se oculta la imagen de la señal de pare
**TiemposReaccion[contador]=Double.Parse(TiempoTranscurrido)TiemposPrueba[co
ntador];** TiemposReaccion[contador]: Es un arreglo que almacena los tiempos de reacción.
Double.Parse(TiempoTranscurrido): Convierte el tiempo transcurrido (almacenado como cadena) a un número de tipo double.

Double.Parse(TiempoTranscurrido) - TiemposPrueba[contador]: Calcula el tiempo de reacción restando el tiempo de la prueba del tiempo transcurrido.

Frena = false;// Reseteo

Detenido = false; Verifica que este el estado del vehículo en detenido y se resetean las banderas.

VerificarMovimiento = false; se verifica el estado del recorrido.

IniciarRecorrido(); llamamos a un método para reiniciar el recorrido.

double tiempos_redondeado Math.Round(Math.Round(TiemposReaccion[contador], 2) - tiempoControl, 2); Redondea el tiempo de reacción a dos decimales y ajusta el tiempo de reacción restando el tiempo de Control y redondeando nuevamente a dos decimales.

resultados.Add(new Tests() { Pruebas = contador + 1, Tiempos = tiempos_redondeado, Velocidad = velocidad, Result = tiempos_redondeado <= TiempoAprobacion ? 1 : 0 });

resultados.Add(...): Añade un nuevo objeto "Tests" a la lista resultados.

Tests: tiene propiedades como Pruebas, Tiempos, Velocidad y Resultados.

Pruebas = contador + 1: Establece el número de la prueba.

Tiempos = tiempos_redondeado: Establece el tiempo redondeado de reacción.

Velocidad = velocidad: Establece la velocidad.

Result = tiempos_redondeado <= TiempoAprobacion ? 1 : 0: Determina si el tiempo redondeado está dentro del tiempo de aprobación (1 si está aprobado, 0 si no).

Pedalera_Control_Form.instance.presenta_resultado()); en esta línea de código se llama a un método y se presenta el resultado en la interfaz del usuario.

PausarTemporizador()); pausamos el temporizador

ReseteoTemporizador()); reseteamos el temporizador

contador++; se incrementa el valor del contador en 1

En resumen, Este fragmento de código se ejecuta cuando el freno está activado (Frena es true) y el vehículo está detenido (Detenido es true). Captura el tiempo de reacción del usuario al frenar el vehículo, oculta la señal de "PARE", reinicia las banderas de estado y el recorrido, y guarda el resultado en una lista de pruebas. Además, presenta el resultado en la interfaz de usuario y maneja el temporizador.

if (contador <= TiemposPrueba.Length && Acelera == true); en esta línea de código se verifica que el valor de contador no exceda la longitud del arreglo "TiemposPrueba". Esto asegura que el contador esté dentro de los límites válidos del arreglo.

Acelera == true: Verifica que la variable Acelera sea true, indicando que el acelerador está siendo presionado.

ReiniciarTemporizador()); Llama a una función para reiniciar el temporizador que mide el tiempo transcurrido mientras el acelerador está siendo presionado.

MoverVehiculo()); llama a una función para mover el vehículo, actualizando la posición del vehículo en la simulación haciendo que avance por la carretera.

En resumen, este fragmento de código se ejecuta para controlar el movimiento del vehículo en la simulación cuando el usuario presiona el acelerador. La condición verifica que el contador no exceda el número de tiempos de prueba y que el acelerador esté presionado.

Figura 1.24: Programación pedalera central

```
if (contador <= TiemposPrueba.Length && Frena == true && Detenido == true)//CAPTURA EL TIEMPO DE REACCION AL FRENAR EL VEHICULO
{
    pbxMuralla.Visible = false;//para ocultar la imagen de la señal pare
    TiemposReaccion[contador] = Double.Parse(TiempoTranscurrido) - TiemposPrueba[contador];//GUARDA EL TIEMPO DE REACCION CUANDO
    Frena = false;// Reseteo
    Detenido = false;
    VerificarMovimiento = false;
    IniciarRecorrido();
    double tiempos_redondeado = Math.Round(Math.Round(TiemposReaccion[contador], 2) - tiempoControl, 2);//Redondeo el tiempo para
    resultados.Add(new Tests() { Pruebas = contador + 1, Tiempos = tiempos_redondeado, Velocidad = velocidad, Result = tiempos_r
    Pedalera_Control_Form.instance.presenta_resultado(); //11
}

PausarTemporizador();
ReseteoTemporizador();
contador++;
}

if (contador <= TiemposPrueba.Length && Acelera == true)//RECORRIDO POR CADA TIEMPO DE PRUEBA ESTABLECIDO
{
    ReiniciarTemporizador();
    MoverVehiculo();
}
```

- Entra en la línea de código y se presenta el resultado.
- Reiniciar temporizador// cuando se presiona el acelerador se mueven el vehículo
- Public void: mover vehículo declaro la función y se coloca la lógica.
- Asignar velocidades a la tabla para que las líneas de la carretera dependiendo de estas velocidades se muevan rápido y lentamente y sea más real la simulación

public void MoverVehiculo() Esta línea declara una función llamada “MoverVehiculo” que no devuelve ningún valor (“void”). Es una función pública (“public”), lo que significa que puede ser accedida desde fuera de la clase en la que está definida.

if (TiemposPrueba[contador] == 2 || TiemposPrueba[contador] == 3) Aquí se verifica si el valor en el arreglo “TiemposPrueba” en la posición especificada por la variable “contador” es igual a 2 o 3. Si es así, se ejecuta el bloque de código dentro de este ‘if’.

Acelera = false; Establece la variable 'Acelera' en falso, lo que indica que el vehículo no está acelerando.

velocidad = 15; // Establece la velocidad en 15, la velocidad estándar o predeterminada del vehículo.

VerificarMovimiento = true; // Establece la variable 'VerificarMovimiento' en verdadero, indicando que se ha verificado el movimiento.

if (TiemposPrueba[contador] == 4) Esta línea verifica si el valor en el arreglo “TiemposPrueba” en la posición especificada por “contador” es igual a 4.

Acelera = false; Establece la variable 'Acelera' en falso, lo que indica que el vehículo no está acelerando.

velocidad = 20; Establece la velocidad en 20.

VerificarMovimiento = true; Establece la variable 'VerificarMovimiento' en verdadero, indicando que se ha verificado el movimiento.

if (TiemposPrueba[contador] == 5 || TiemposPrueba[contador] == 6) Verifica si el valor en “TiemposPrueba” está en la posición indicada por el “contador” igual a 5 o 6.

Acelera = false; Establece la variable “Acelera” en falso, lo que indica que el vehículo no está acelerando.

velocidad = 30; Establece la velocidad en 30.

VerificarMovimiento = true; Establece la variable 'VerificarMovimiento' en verdadero, indicando que se ha verificado el movimiento.

private void IniciarRecorrido() Función que inicializa el tiempo en cero y comienza el conteo de segundos para el cronómetro.

tiempo_segundo = 0; Establece la variable 'tiempo_segundo' en cero, iniciando así el cronómetro.

private void PararRecorrido() Establece la variable de tiempo en cero y hace visible la señal de pare.

tiempo_segundo = 0; Establece la variable 'tiempo_segundo' en cero, deteniendo el cronómetro.

pbxMuralla.Visible = true; Hace visible la señal de pare (representada por el control 'pbxMuralla').

public void DetenerVehiculo() Función que detiene el vehículo, detiene el cronómetro y lo reinicia para detener las líneas.

if (VerificarMovimiento == true) Verifica si el vehículo está en movimiento.

SwTiempoReaccion.Stop(); Detiene el cronómetro.

IniciarRecorrido(); Reinicia el cronómetro para detener las líneas.

private void timer1_Tick(object sender, EventArgs e) Este método se llama cada vez que el temporizador 'timer1' dispara un evento Tick de esta manera se actualiza el estado de la animación del simulador.

RealizarPrueba(); Llama al método “RealizarPrueba”, que realiza una acción relacionada con la ejecución de una prueba.

private void Pedalera_Test_KeyDown(object sender, KeyEventArgs e) Este método maneja el evento KeyDown que es la encargada de captar las señales del teclado en el control Pedalera_Test.

//RecibirDatosTeclado("A"); Llama a una función para procesar los datos de teclado con la tecla "A" (para simular una acción).

if (e.KeyCode == Keys.A) Verifica si la tecla presionada es la tecla "A".

RecibirDatosTeclado("A"); Llama a la función “RecibirDatosTeclado()” con el argumento "A".

if (e.KeyCode == Keys.D) Verifica si la tecla presionada es la tecla "D".

RecibirDatosTeclado("B"); Llama a la función “RecibirDatosTeclado()” con el argumento "B"

public void RecibirDatosTeclado(String strData) Esta función recibe datos del teclado para identificar si se presiona el acelerador o el freno.

//DatosPuertoSerial = serialPort1.ReadLine(); Lee datos del puerto serial (comentado).

DatosPuertoSerial = strData; Asigna los datos recibidos del teclado a la variable DatosPuertoSerial.

DelegarAccionPedal EscribirAccionPedal = new DelegarAccionPedal (VerificarActionPedal); Crea una instancia de “DelegarAccionPedal” y la inicializa con el método VerificarActionPedal.

this.Invoke(EscribirAccionPedal); Se involucra el subproceso que posee el control.

public void VerificarActionPedal() Esta función implementa la lógica para cada pulso de acción del pedal.

if (DatosPuertoSerial == "B" && bandera_frena) Verifica si los datos del puerto serial son "B" y la bandera de freno está activada.

FrenarVehiculo(); Llama al método “FrenarVehiculo” para aplicar el freno del vehículo.

bandera_frena = false; Desactiva la bandera o pulso de freno.

bandera_acelera = true; Activa la bandera o pulso de aceleración.

bandera_noaplasta = false; Desactiva la bandera o pulso de no aplastar.

if (DatosPuertoSerial == "A" && bandera_acelera) Verifica si los datos del puerto serial son "A" y la bandera de aceleración está activada.

AcelerarVehiculo(); Llama al método “AcelerarVehiculo” para aumentar la velocidad del simulador.

bandera_acelera = false; Desactiva la bandera de aceleración.

bandera_frena = true; Activa la bandera de freno.

bandera_noaplasta = true; Activa la bandera de no aplastar.

bandera_habilita = true; Activa la bandera de habilitación.

if (DatosPuertoSerial == "C" && bandera_noaplasta) // Verifica si los datos del puerto serial son "C" y la bandera de no aplastar está activada.

DetenerVehiculo(); Llama al método DetenerVehiculo para detener el vehículo.

bandera_acelera = true; Activa la bandera de aceleración.

bandera_frena = true; Activa la bandera de freno.

bandera_noaplasta = false; Desactiva la bandera de no aplastar.

public class Tests : IEquatable<Tests> Declaración de la clase Tests, que implementa la interfaz IEquatable<Tests>.

public int Pruebas { get; set; } Propiedad que representa el número de pruebas.

public int Velocidad { get; set; } Propiedad que representa la velocidad.

public double Tiempos { get; set; } Propiedad que representa el tiempo.

public int Result { get; set; } Propiedad que representa el resultado.

public override bool Equals(object obj) Sobrescribe el método Equals para comparar objetos.

if (obj == null) Verifica si el objeto proporcionado es nulo. **return false;** Devuelve falso si el objeto es nulo.

Tests objAsTests = obj as Tests; Convierte el objeto proporcionado a tipo Tests.

if (objAsTests == null) Verifica si la conversión fue exitosa.

return false; Devuelve falso si la conversión falla.

else return Equals(objAsTests); Llama al método Equals(objAsTests) para comparar el objeto actual con el objeto convertido.

public override int GetHashCode() Sobrescribe el método GetHashCode para generar un código hash.

return Velocidad; Devuelve el código hash basado en la propiedad Velocidad.

public bool Equals(Tests other) Implementa el método Equals para comparar dos objetos Tests.

if (other == null) Verifica si el objeto proporcionado es nulo. **return false;** Devuelve falso si el objeto es nulo.

return (this.Pruebas.Equals(other.Pruebas)); Compara las propiedades Pruebas de los dos objetos y devuelve el resultado de la comparación.

public bool Equals2(Tests other) // Implementa otro método Equals para comparar dos objetos Tests basados en la propiedad Velocidad.

if (other == null) Verifica si el objeto proporcionado es nulo. **return false;** Devuelve falso si el objeto es nulo.

return (this.Velocidad.Equals(other.Velocidad)); // Compara las propiedades Velocidad de los dos objetos y devuelve el resultado de la comparación.

Figura 1.25: Programación lista de resultados.

```
public class Tests : IEquatable<Tests> //Clase
{
    4 referencias
    public int Pruebas { get; set; }
    5 referencias
    public int Velocidad { get; set; }
    2 referencias
    public double Tiempos { get; set; }

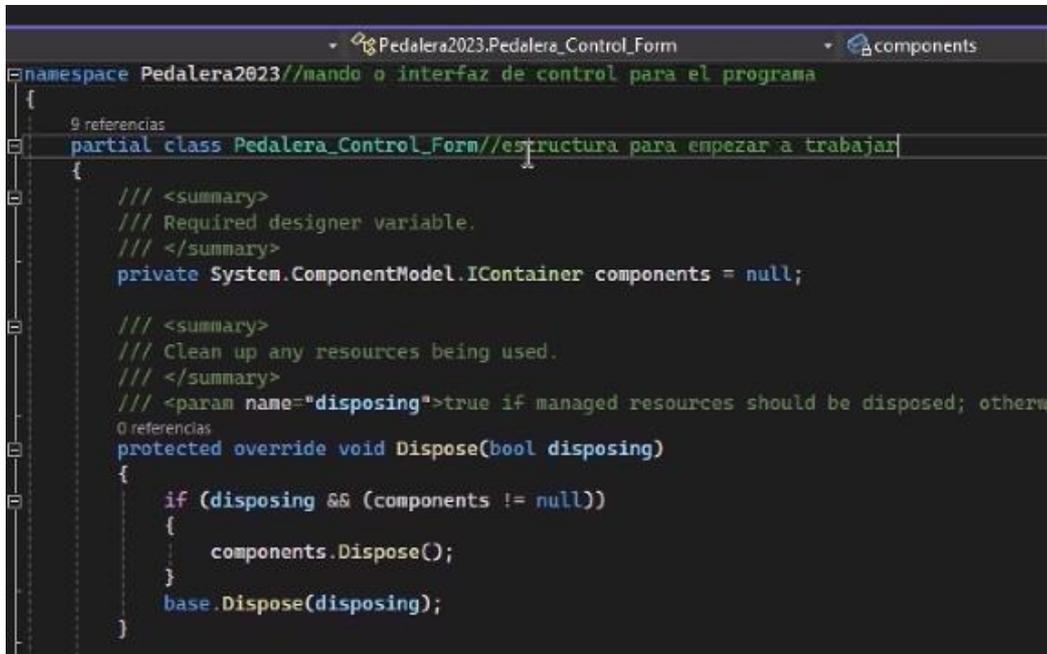
    2 referencias
    public int Result { get; set; }

    0 referencias
    public override bool Equals(object obj)
    {
        if (obj == null) return false;
        Tests objAsTests = obj as Tests;
        if (objAsTests == null) return false;
        else return Equals(objAsTests);
    }
}
```

Esta clase se utiliza para declarar la lista de resultados donde se almacenan los parámetros de las pruebas de velocidad y tiempo. En esta lista se guardan los datos relevantes de cada prueba, como la velocidad alcanzada y el tiempo empleado. Esencialmente, esta clase proporciona una estructura organizada para almacenar y gestionar los resultados obtenidos durante las pruebas, lo que facilita el análisis y seguimiento de los datos de rendimiento del vehículo

Control Form Pedalera C#

Figura 1.26: Programación public class



```
namespace Pedalera2023 //mando o interfaz de control para el programa
{
    9 referencias
    partial class Pedalera_Control_Form //estructura para empezar a trabajar
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        0 referencias
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

TEST RESISTENCIA A LA MONOTONIA

En el siguiente apartado, se presentan las líneas de programación del test de resistencia a la monotonía desarrollado para el estudio dentro del marco de la tesis. Este test constituye una herramienta crucial para evaluar cómo los individuos responden y mantienen su atención frente a estímulos repetitivos y predecibles. A través de la descripción detallada de su estructura y funcionalidad, se podrá entender mejor cómo se aplicará en el contexto de la investigación, contribuyendo así al análisis integral de la capacidad humana de resistir la monotonía en entornos específicos.

Programación:

public partial class Resistencia : Form Declaración de la clase Resistencia, que se hereda de la clase Form.

public static sbyte hours = 0, minutes = 0, tempSeconds1, tempSeconds2; // Declaración de variables estáticas para horas, minutos y segundos temporales.

private int seconds = 0, bandera_imagen = 0, bandera_freno = 0, bandera_abajo = 0, bandera_arriba = 0, bandera_acelera = 0, secondstemp = 0, contador_prueba = 0, numero_de_pruebas = 9; Declaración de variables privadas para segundos, banderas de imagen, freno, abajo, arriba, aceleración, segundos temporales, contador de pruebas y número de pruebas.

private string DatosPuertoSerial; Declaración de una variable privada para almacenar datos del puerto serial.

public delegate void DelegarAccionPedal(); Declaración de un delegado llamado DelegarAccionPedal.

private void btnFrena_Click(object sender, EventArgs e) Método que se ejecuta cuando se hace clic en el botón de freno.

frenar(); // Llama al método frenar().

private void cumple_tiempo_botones() Método para verificar el tiempo y qué botón se ha presionado.

int resultado = 0; Variable para almacenar el resultado.

if (seconds <= 5) Verifica si los segundos son menores o iguales a 5.

resultado = 1; Si es así, el resultado es 1, lo que significa que se cumplió el tiempo.

if (seconds > 5) Verifica si los segundos son mayores a 5.

resultado = 2; Si es así, el resultado es 2, lo que significa que no se cumplió el tiempo.

tests.Add(new Tests() { Pruebas = contador_prueba, Tiempo = seconds, Resultado_tiempo = resultado, Resultado_botones = 1 }); Agrega un nuevo objeto Tests a la lista tests con los datos de la prueba actual.

TestResistenciaControlForm.instance.presenta_resultado(); Llama al método presenta_resultado() de la instancia.

private void no_cumple_botones() Método para cuando no se cumple con los botones.

int resultado = 0; Variable para almacenar el resultado.

if (seconds <= 5) Verifica si los segundos son menores o iguales a 5.

resultado = 1; Si es así, el resultado es 1, lo que significa que se cumplió el tiempo.

if (seconds > 5) Verifica si los segundos son mayores a 5.

resultado = 2; Si es así, el resultado es 2, lo que significa que no se cumplió el tiempo.

tests.Add(new Tests() { Pruebas = contador_prueba, Tiempo = seconds, Resultado_tiempo = resultado, Resultado_botones = 2 }); Agrega un nuevo objeto Tests a la lista tests con los datos de la prueba actual.

TestResistenciaControlForm.instance.presenta_resultado(); Llama al método `presenta_resultado()` de la instancia.

public void frenar() Método para modificar las variables según el botón de freno.

acelera = false; Establece la variable `acelera` en falso.

frena = true; Establece la variable `frena` en verdadero, indicando que se está frenando.

arriba = false; Establece la variable `arriba` en falso.

abajo = false; Establece la variable `abajo` en falso.

private void btnAcelera_Click(object sender, EventArgs e) Método que se ejecuta cuando se hace clic en el botón de acelerar.

acelerar(); Llama al método `acelerar()`.

public void acelerar() Método para modificar las variables según el botón de acelerar.

acelera = true; Establece la variable `acelera` en verdadero, indicando que se está acelerando.

frena = false; Establece la variable `frena` en falso.

arriba = false; Establece la variable `arriba` en falso.

abajo = false; Establece la variable `abajo` en falso.

private void timer1_Tick(object sender, EventArgs e) Método que se llama cada vez que el temporizador 'timer1' dispara un evento Tick.

if (contador_prueba < numero_de_pruebas) Verifica si el contador de pruebas es menor que el número de pruebas.

switch (bandera_imagen) Evalúa la variable bandera_imagen.

case 0: Si bandera_imagen es 0, muestra la imagen de la señal de "no pase".

pbxSeñal.Image = Properties.Resources.no_pase;

break;

case 1: Si bandera_imagen es 1, muestra la imagen de la señal de "ceda".

pbxSeñal.Image = Properties.Resources.ceda;

break;

case 2: Si bandera_imagen es 2, muestra la imagen de la señal de "cruce".

pbxSeñal.Image = Properties.Resources.cruce;

break;

case 3: Si bandera_imagen es 3, muestra la imagen de la señal de "semaforo".

pbxSeñal.Image = Properties.Resources.semaforo;

break;

lblsegundos.Text = tempSeconds1.ToString(); Actualiza el texto del control lblsegundos con el valor de tempSeconds1 convertido a cadena.

if (tempSeconds1 != (sbyte)DateTime.Now.Second && inicio) Comprueba si tempSeconds1 es diferente al segundo actual y si la variable inicio es verdadera.

seconds++; Incrementa el contador de segundos.

tempSeconds1 = (sbyte)DateTime.Now.Second; Actualiza tempSeconds1 con el segundo actual.

lblSeconds.Text = seconds.ToString(); Actualiza el texto del control lblSeconds con el valor de la variable seconds convertido a cadena.

if (bandera_imagen == 0) Verifica si la bandera_imagen es 0.

if (frena) Verifica si la variable frena es verdadera.

switch (bandera_freno) Evalúa la variable bandera_freno.

case 0: Si bandera_freno es 0, muestra un mensaje, llama a la función cumple_tiempo_botones() y actualiza la bandera_freno.

```
lblMensaje.Text = "¡BIEN!";
```

```
cumple_tiempo_botones();
```

```
bandera_freno = 1;
```

```
break;
```

case 1: // Si bandera_freno es 1, establece secondstemp y actualiza la bandera_freno.

```
secondstemp = seconds;
```

```
bandera_freno = 2;
```

```
break;
```

case 2: Si bandera_freno es 2, verifica si han pasado 2 segundos, reinicia los valores y actualiza contador_prueba.

```
if (seconds == secondstemp + 2)
```

```
seconds = 0;
```

```
frena = false;
```

```
acelera = false;
```

```
abajo = false;
```

```
arriba = false;
```

```
contador_prueba++;
```

```
cambaiimagen();
```

else if (abajo || arriba || acelera) Verifica si abajo, arriba o acelera son verdaderos.

switch (bandera_freno) Evalúa la variable bandera_freno.

case 0: Si bandera_freno es 0, muestra un mensaje, llama a la función "no_cumple_botones()" y actualiza la bandera_freno.

```
lblMensaje.Text = "¡MAL!";
```

```
no_cumple_botones();
```

```
bandera_freno = 1;
```

```
break;
```

case 1: Si bandera_freno es 1, establece secondstemp y actualiza la bandera_freno.

```
secondstemp = seconds;
```

```
bandera_freno = 2;
```

break;

case 2: Si “bandera_freno” es 2, verifica si han pasado 2 segundos, reinicia los valores y actualiza “contador_prueba.”

if (seconds == secondstemp + 2)

seconds = 0;

frena = false;

acelera = false;

abajo = false;

arriba = false;

contador_prueba++;

cambiaimagen();

lblMensaje.Text = "Prueba Finalizada"; Establece el texto del control lblMensaje como "Prueba Finalizada".

lblMensaje.Location = new System.Drawing.Point(515 - 300, 480); Establece la ubicación del control lblMensaje.

public void cambiaimagen() Método para cambiar la imagen dependiendo de la prueba.

Random r = new Random(); Se crea una instancia de la clase Random para generar números aleatorios.

int banderatemp = 0; Variable para almacenar temporalmente un valor aleatorio.

bandera_imagen++; Incrementa el valor de la variable “bandera_imagen”.

if (bandera_imagen >= 4) Verifica si “bandera_imagen” es mayor o igual a 4.

bandera_imagen = 0; Si es así, reinicia “bandera_imagen” a 0.

lblMensaje.Text = ""; Establece el texto del control “lblMensaje” como una cadena vacía.

bandera_abajo = bandera_acelera = bandera_arriba = bandera_freno = 0; Reinicia las banderas relacionadas a los botones.

private void btnDerecha_Click(object sender, EventArgs e) Método que se ejecuta cuando se hace clic en el botón de la derecha.

Derecha(); Llama al método Derecha().

public void Derecha() Método para establecer las variables al presionar el botón derecho.

acelera = false; Establece la variable acelera en falso.

frena = false; Establece la variable frena en falso.

arriba = true; Establece la variable arriba en verdadero, indicando que se está pulsando hacia arriba.

abajo = false; Establece la variable abajo en falso.

private void btnIzquierda_Click(object sender, EventArgs e) Método que se ejecuta cuando se hace clic en el botón de izquierda.

Izquierda(); Llama al método Izquierda().

public void Izquierda() Método para establecer las variables al presionar el botón de izquierda.

acelera = false; Establece la variable acelera en falso.

frena = false; Establece la variable frena en falso.

arriba = false; Establece la variable arriba en falso.

abajo = true; Establece la variable abajo en verdadero, indicando que se está pulsando hacia abajo.

private void button3_Click(object sender, EventArgs e) Método que se ejecuta cuando se hace clic en el botón button3.

resetear(); Llama al método resetear().

public void resetear() Método para reiniciar todas las variables y preparar para un nuevo test.

tests.Clear(); Borra todos los elementos de la lista "tests".

inicio = false; Establece la variable inicio en falso.

acelera = frena = arriba = abajo = false; Establece las variables acelera, frena, arriba y abajo en falso.

lblMensaje.Text = ""; Establece el texto del control lblMensaje como una cadena vacía.

bandera_abajo = bandera_acelera = bandera_arriba = bandera_freno = bandera_imagen = 0; Reinicia las variables de bandera.

lblMensaje.Location = new Point(515, 480); // Establece la ubicación del control lblMensaje.

contador_prueba = 0; Reinicia el contador de prueba.

pbxSeñal.Visible = false; Oculta el control pbxSeñal.

label1.Visible = true; Hace visible el control label1.

private bool acelera = false, frena = false, arriba = false, abajo = false, inicio = false;

Declaración de variables booleanas.

public List<Tests> tests = new List<Tests>(); Declaración de una lista de tipo Tests.

public Resisgencia() Constructor de la clase Resisgencia.

InitializeComponent(); Inicializa los componentes del formulario.

timer1.Start(); Inicia el temporizador timer1.

Bounds = Screen.PrimaryScreen.Bounds; Establece los límites del formulario a los límites de la pantalla primaria.

FormBorderStyle = FormBorderStyle.None; Establece el estilo de borde del formulario como None, es decir, sin bordes.

private void button4_Click(object sender, EventArgs e) Método que se ejecuta cuando se hace clic en el botón button4.

iniciar(); Llama al método iniciar().

private void Resisgencia_KeyDown(object sender, KeyEventArgs e) Método que se ejecuta cuando se presiona una tecla mientras el formulario tiene el foco.

if (e.KeyCode == Keys.A) Verifica si la tecla presionada es la tecla A.

RecibirDatosTeclado("A"); Llama al método RecibirDatosTeclado con el argumento "A".

if (e.KeyCode == Keys.D) Verifica si la tecla presionada es la tecla D.

RecibirDatosTeclado("B"); Llama al método RecibirDatosTeclado con el argumento "B".

if (e.KeyCode == Keys.Q) Verifica si la tecla presionada es la tecla Q.

RecibirDatosTeclado("C"); Llama al método RecibirDatosTeclado con el argumento "C".

if (e.KeyCode == Keys.W) Verifica si la tecla presionada es la tecla W.

RecibirDatosTeclado("D"); Llama al método RecibirDatosTeclado con el argumento "D".

public void RecibirDatosTeclado(String strData) Método para recibir datos de teclado y enviar la lógica correspondiente.

DatosPuertoSerial = strData; Asigna el valor recibido al campo DatosPuertoSerial.

DelegarAccionPedal EscribirAccionPedal = new DelegarAccionPedal (VerificarAccionesPedal); Crea una instancia de la delegado DelegarAccionPedal y la inicializa con el método VerificarAccionesPedal.

this.Invoke(EscribirAccionPedal); // Invoca el método VerificarAccionesPedal a través de la instancia de delegado, utilizando la sincronización del hilo que posee este control (ya que parece ser un método de un formulario).

public void VerificarAccionesPedal() Método para verificar las acciones del pedal y llamar a los métodos correspondientes.

if (DatosPuertoSerial == "B") Verifica si se recibió el dato "B" (indicando que se aplastó el freno).

frenar(); Llama al método frenar().

if (DatosPuertoSerial == "A") Verifica si se recibió el dato "A" (indicando que se aplastó el acelerador).

acelerar(); Llama al método acelerar().

if (DatosPuertoSerial == "C") Verifica si se recibió el dato "C" (indicando que se aplastó el pedal derecho).

Derecha(); Llama al método Derecha().

if (DatosPuertoSerial == "D") Verifica si se recibió el dato "D" (indicando que se aplastó el pedal izquierdo).

Izquierda(); Llama al método Izquierda().

PROGRAMACION DEL SIMULADOR DE CONDUCCION 'TEST DE RESISTENCIA A LA MONOTONIA'

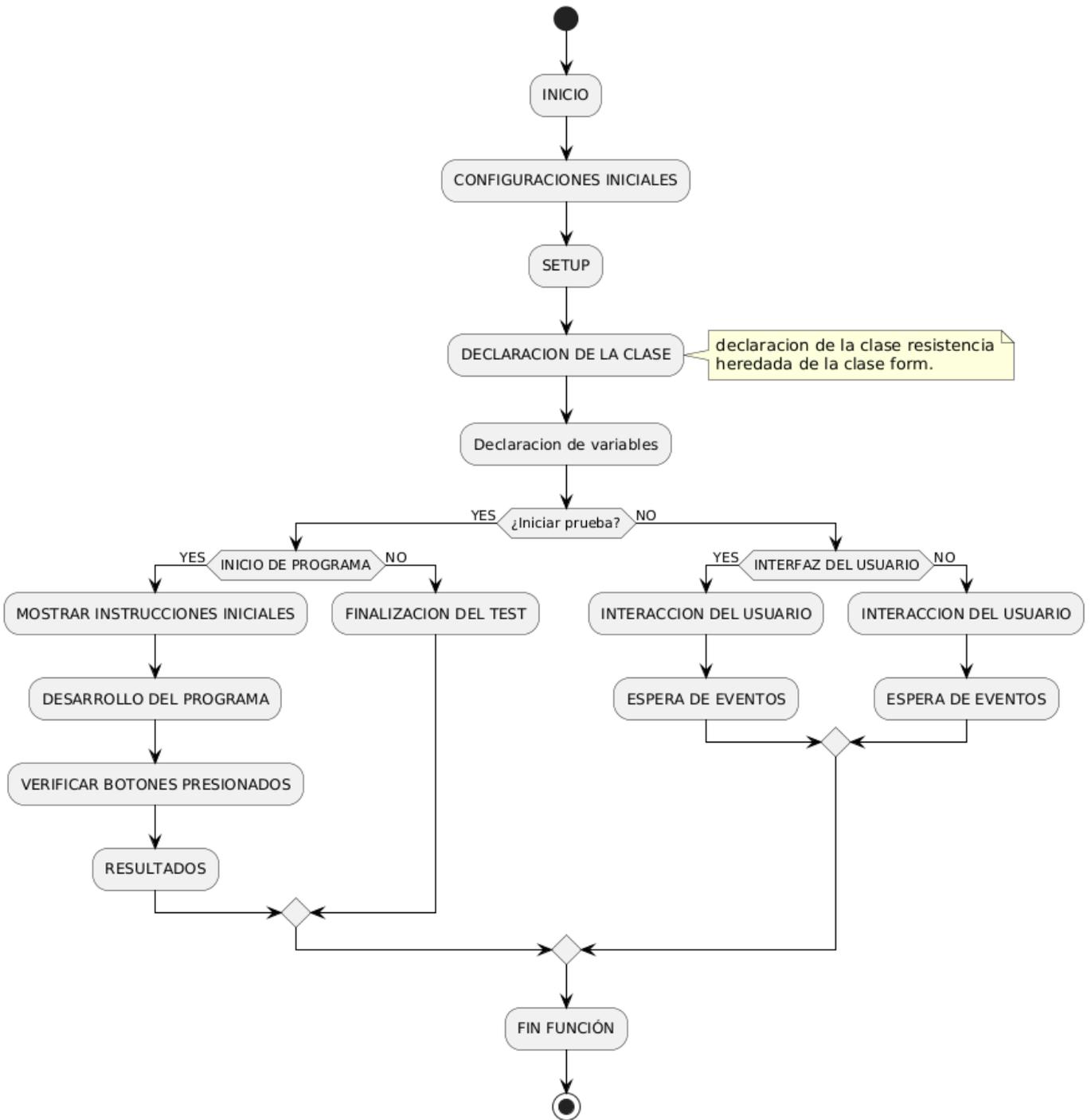


Figura 1.27: Programación pulsos de los pedales.

```
- Pedalera2023,Resistencia
}
else if (abajo || arriba || acelera)
{
    switch (bandera_freno)
    {
        case 0:
            lblMensaje.Text = "¡MAL!";
            no_cumple_botones();
            bandera_freno = 1;
            break;
        case 1:
            seconds = void Resistencia.no_cumple_botones();
            bandera_freno = 2;
            break;
        case 2:
            if (seconds == secondstemp + 2)
            {
                seconds = 0;
                frena = false;
                acelera = false;
                abajo = false;
                arriba = false;
                contador_prueba++;
                cambiaimagen();

                //bandera_freno = 3;
            }
    }
}
```

Esta línea de código ajusta la variable según el botón activado. Si se presiona el botón correcto, se establece el mensaje "¡bien!".

Se propone el caso en el que el botón se coloca en mal si no cumple con los parámetros.

Hay 4 casos que se repiten de forma cíclica para este test.

Figura 1.28: Declaración de las directivas.

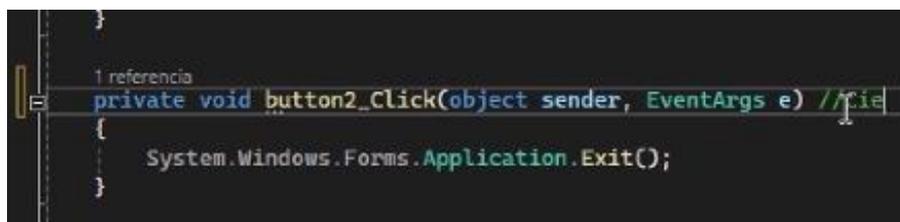
```
1 referencia
private void btnStart_Click_2(object sender, EventArgs e) //Llamo la funcion iniciar del otro form
{
    _testResistencia.iniciar();
}

1 referencia
private void button1_Click_1(object sender, EventArgs e) //Llamo la funcion reset del otro forms
{
    _testResistencia.resetear();
    numero_aprobados = 0;
}

1 referencia
private void button2_Click(object sender, EventArgs e) //Cierro la aplicacion
{
    System.Windows.Forms.Application.Exit();
}
```

Esta función me proporciona la capacidad de, al presionar el botón de prueba de reacción simple, separar la prueba de reacción del control de la pedalera y especificar en qué pantalla proyectar cada formulario.

Figura 1.29: Cierre del programa.



```
1 referencia
private void button2_Click(object sender, EventArgs e) //Clic
{
    System.Windows.Forms.Application.Exit();
}
```

1.5. Test de resistencia a la monotonía.

El test de resistencia a la monotonía brinda información sobre la capacidad de concentración que tiene el usuario, asociando los recuerdos recientes con números presentados de manera monótona y repetitiva, con este ejercicio se logra la fatiga y fragmentación de la atención. Para ello se muestran 4 números relacionados con 4 acciones (reacciones) a realizar por el usuario, mostradas siempre en la misma secuencia en un transcurso de dos minutos, lo que provoca fatiga por la repetición y monotonía por la secuencia continua (Biartic, 2022b).

El objetivo de evaluar la capacidad de una persona para completar un viaje largo sin perder la concentración. Esto se logra mediante un test que se presenta de forma secuencial en un orden fijo con muchas repeticiones donde se requiere que la persona responda de alguna manera a cada uno de estos estímulos. Se debe indicar al usuario que utilice los pedales y los botones izquierdo y derecho, luego se procede a hacer clic en el botón "hacer una prueba". Cuando comienza el test, se muestra el cronograma y la ayuda a la derecha le indica qué hacer. (Decreto 769 Del Nuevo Código Nacional de Tránsito Terrestre.)

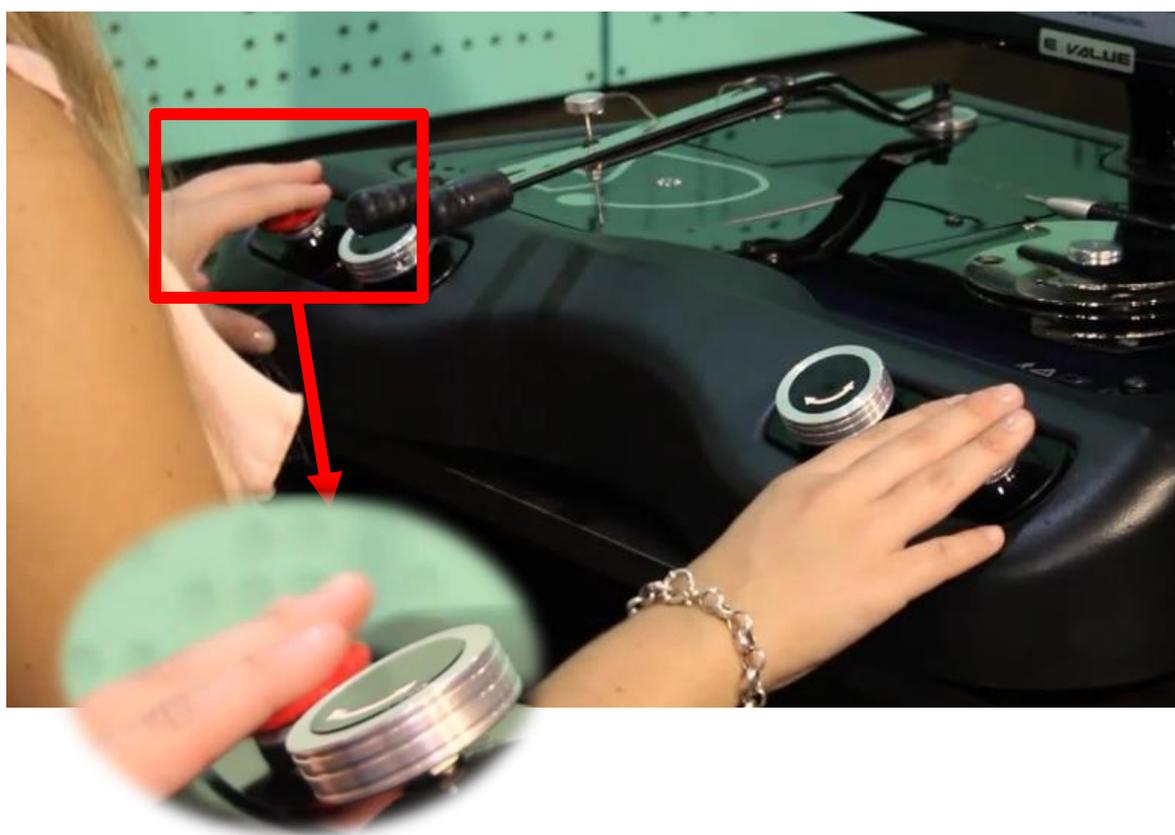
La asistencia solo se brindará con los primeros cuatro estímulos a medida que se practican, por lo que los evaluados deben recordar cada estímulo y asociarlo con la acción dada. Si no hace esto durante la fase de práctica, aparecerá un mensaje de error en la parte inferior de la pantalla. Nuevamente aparecerá el mismo gráfico y se le pedirá que realice la acción correcta indefinidamente hasta que lo haga. Pero si el evaluado lo hace durante la evaluación, pasará a la siguiente imagen. También se muestra un mensaje cuando la operación se realiza correctamente, que en todos los casos lleva a la imagen siguiente. Después de la fase de prueba, la ayuda en el lado derecho de la imagen ya no aparece. Muestra el número de

iteraciones restantes bajo el botón "Incompleto" y detalles sobre cada reacción a continuación. Cuando se complete el examen, los resultados se mostrarán en la pantalla de vista previa con la selección de resultados habilitada (se muestran los resultados recomendados), el campo de monitoreo y el botón Guardar examen (Biartic, 2022c).

Después de ingresar la información relevante, continuamos haciendo clic en el botón "Guardar".

Se considera que el tiempo medio de respuesta está dentro del rango normal si el tiempo medio de respuesta es inferior a 0,63 segundos para licencias profesionales, menos de 0,53 segundos para licencias particulares y menos de 0,43 segundos para licencias generales. Si alguna de las escalas involucradas excede los valores normales, se recomienda un resultado "Fallo". (Resolución 1555 del Ministerio de transporte)

Figura 1.30: Botonera para el Test



Fuente: (Biartic, 2022c).

Figura 1.31: Pantalla del Test de resistencia a la monotonía



Fuente: (Biartic, 2022c).

Figura 1.32: Control del test a la resistencia a la monotonía

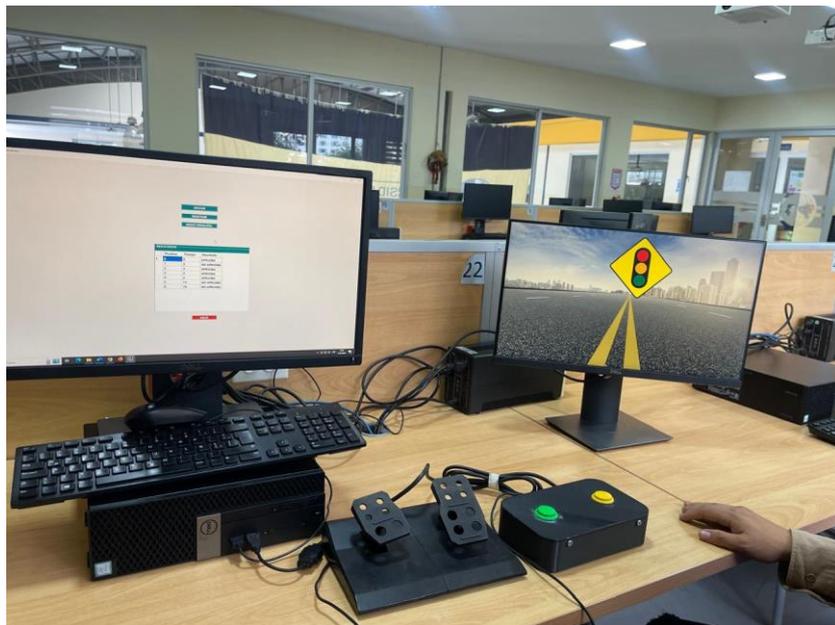
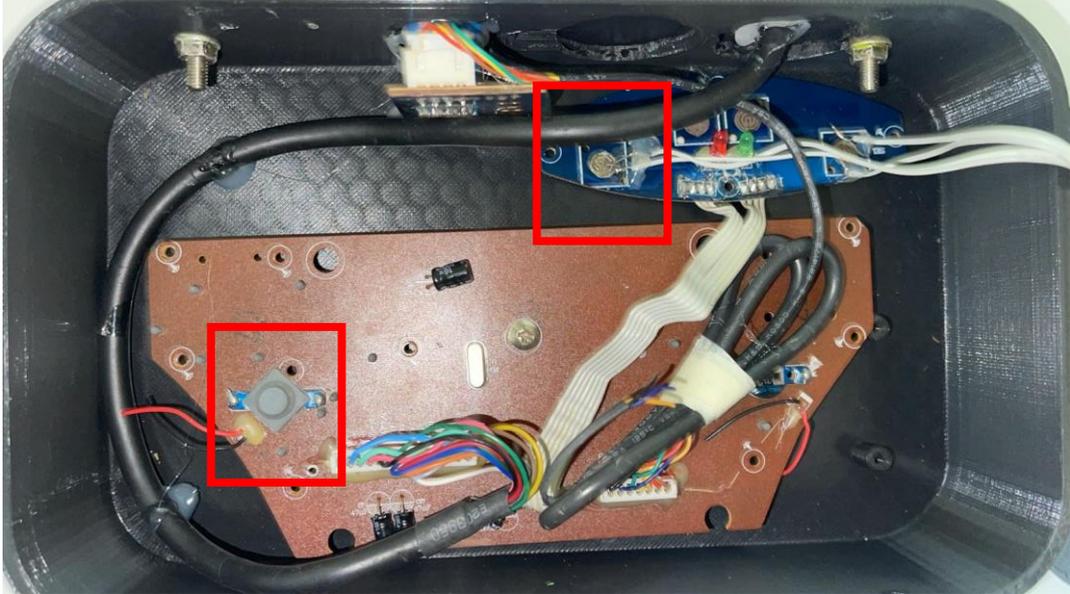


Figura 1.33: Botonera del simulador de conducción



La placa está diseñada para facilitar la realización del test, ya que incluye pulsadores que ayudan a evaluar la concentración de los conductores mientras viajan en carreteras largas. El simulador está completamente preparado para su uso: en el monitor se muestra el test en sí, mientras que en la laptop se registran y analizan los resultados obtenidos. Esto permite recabar datos del test y determinar si el conductor mantiene la concentración adecuada y reacciona correctamente ante señales de pare durante el trayecto.

Figura 1.34: Impresión 3D botonera parte superior

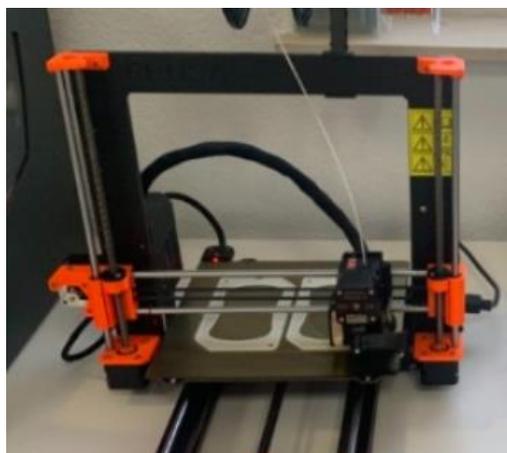


Figura 1.35: Impresión 3D botonera parte inferior

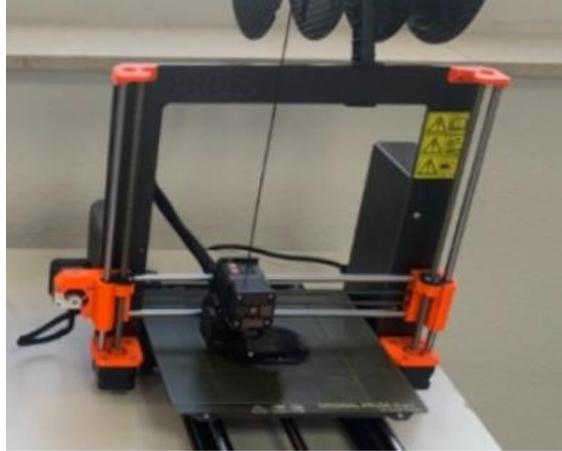
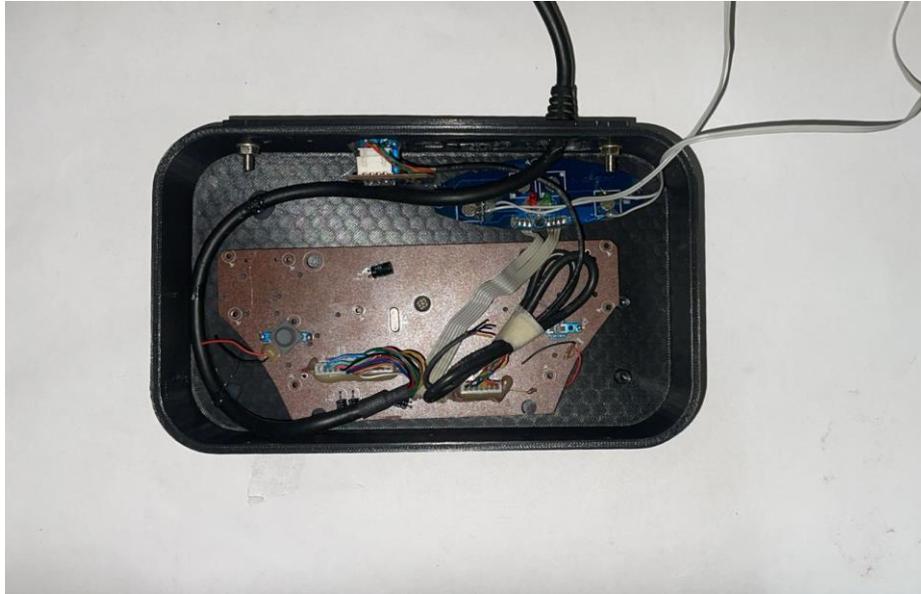


Figura 1.36: Mecanismo de la botonera.



Botonera que consta del cerebro sacado del volante de pruebas. (se sacó el cerebro del volante y se adaptó a una caja impresa en 3d).

Figura 1.37: Adaptación de la placa a la botonera



El proceso de integración del sistema de control del volante se insertó en una caja conocida como "botonera" "fabricada en 3D", fue realizado con meticulosa precisión para asegurar tanto su funcionalidad como su estética en el simulador. Inicialmente, se diseñó la botonera utilizando software de modelado 3D, garantizando que cumpliera con las especificaciones y dimensiones adecuadas para alojar de manera óptima el sistema de control del volante. Al realizar una prueba antes de comenzar con los participantes se detectó lo siguiente:

Figura 1.38: Verificación del pedal (antes)

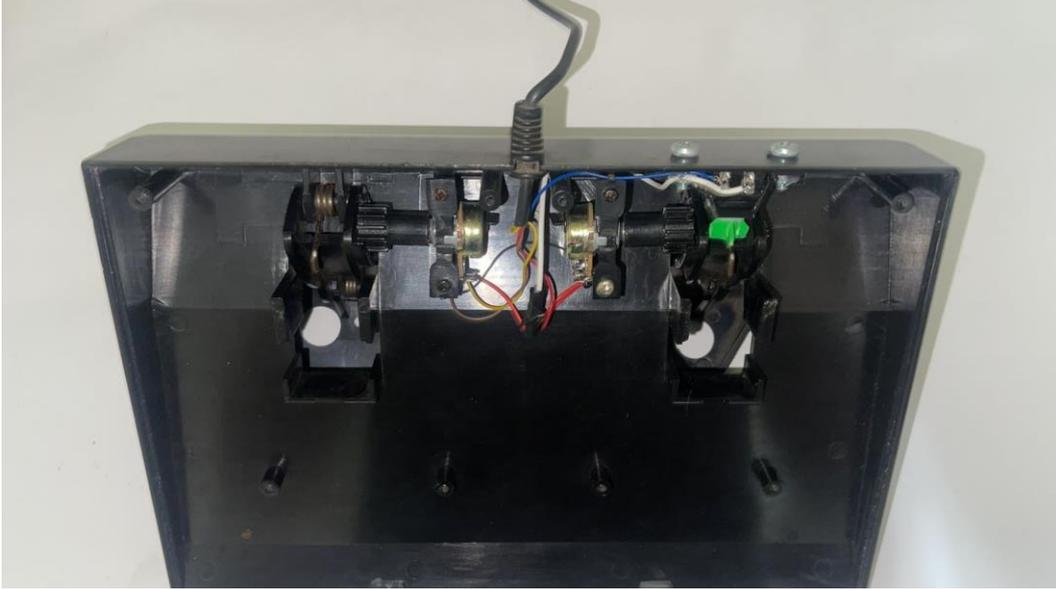


Figura 1.39: Verificación de la pedalera (después)

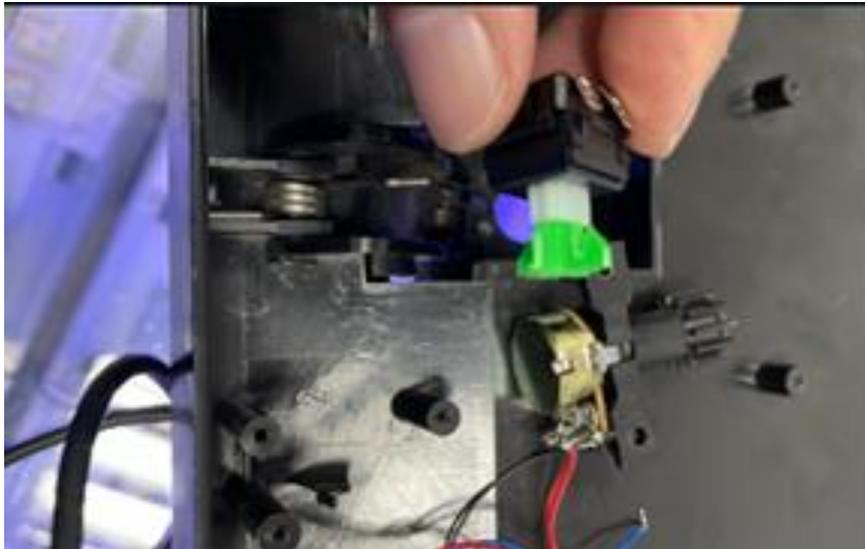
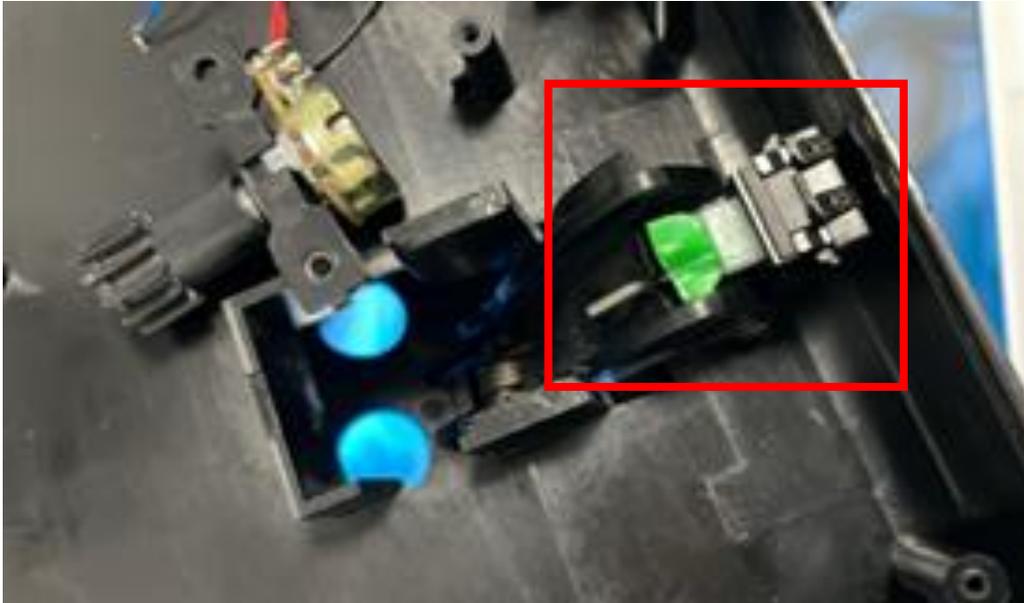


Figura 1.40: Pulsador adaptado al pedal



Se ha instalado un pulsador en la pedalera, ubicado en la parte del pedal, con el fin de mejorar la precisión de los resultados durante la ejecución del test.

Figura 1.41: Presentación del test de reacción simple

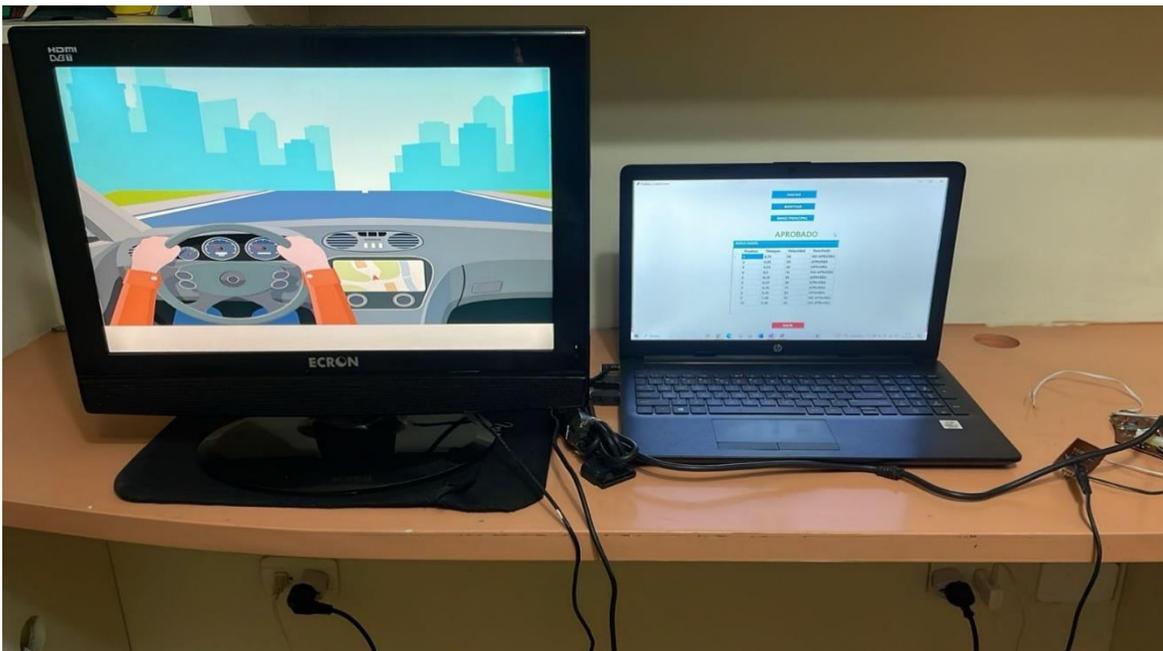


Figura 1.42: Presentación del test de Resistencia a la monotonía

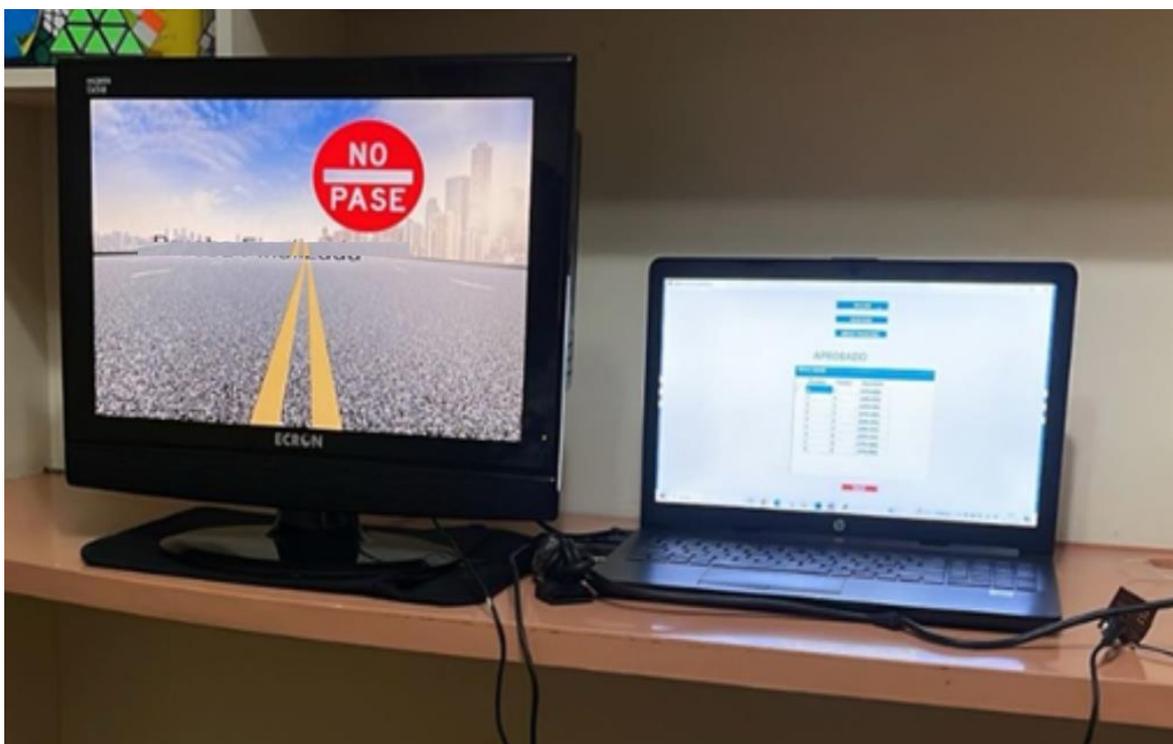


Figura 1.43: Aplicación del Test de reacción simple

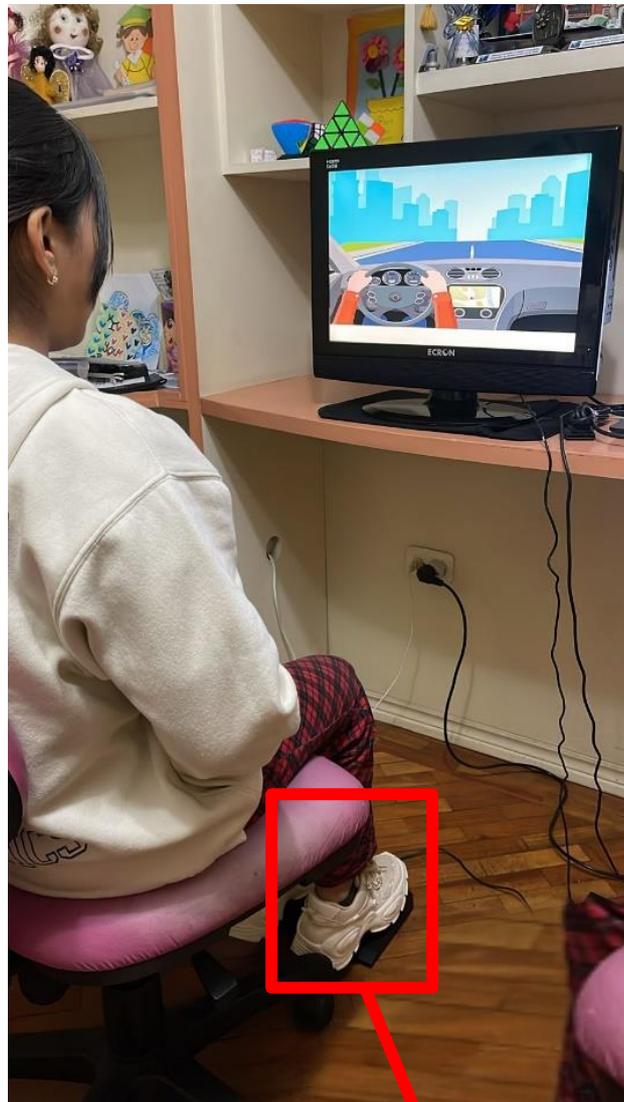


Figura 1.44: Finalización del test de reacción simple

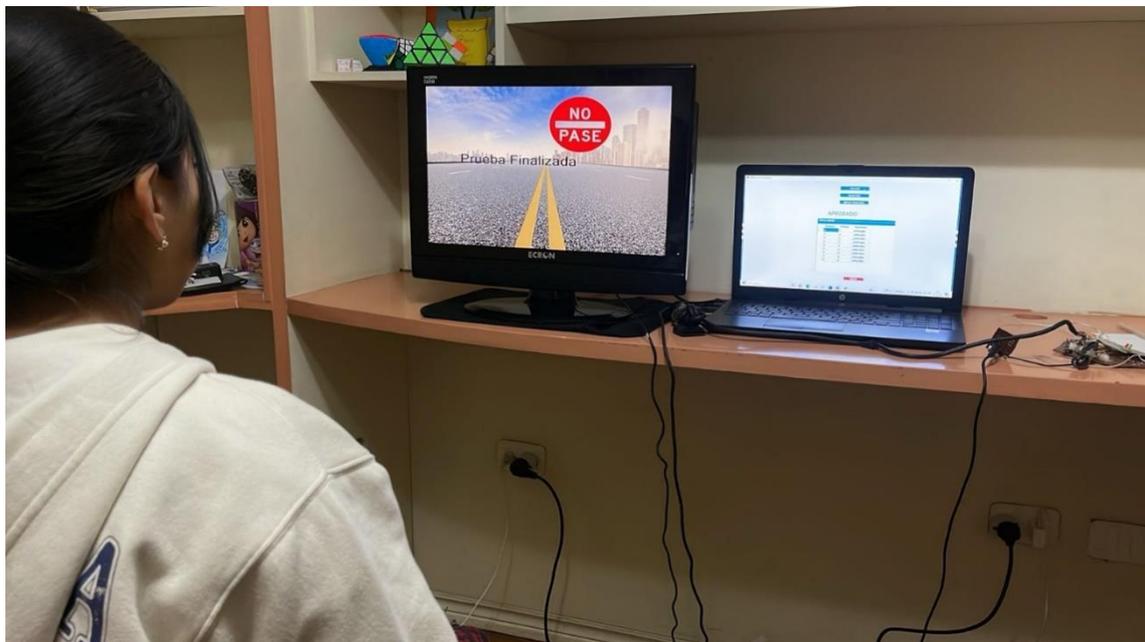
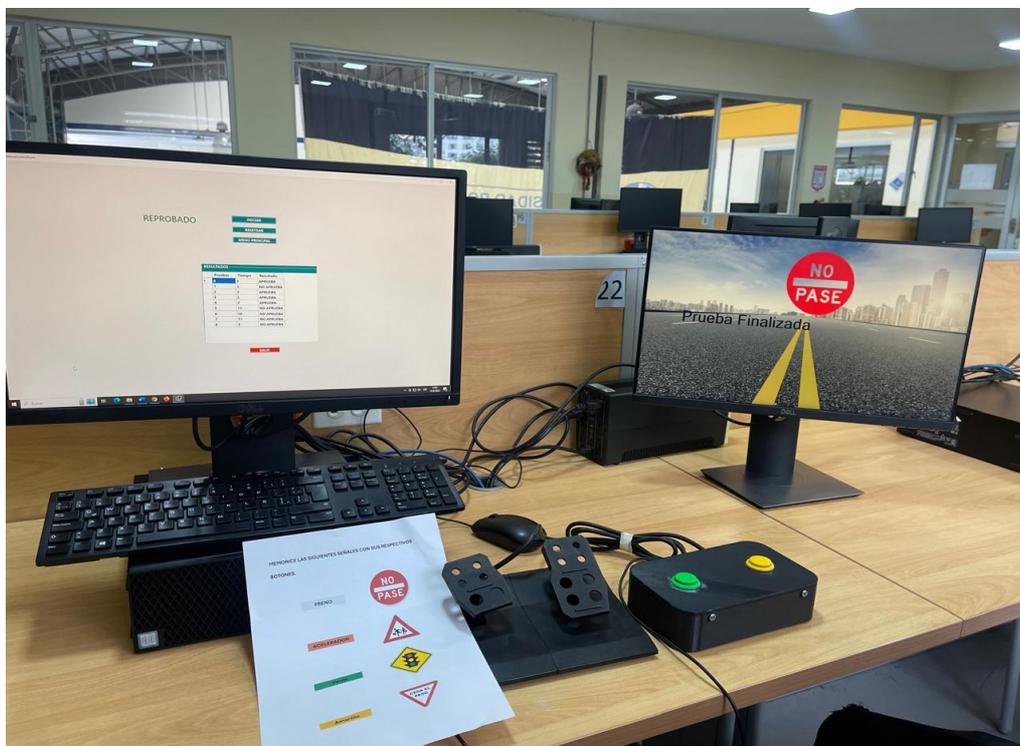


Figura 1.45: Presentación final



CAPÍTULO 2

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS OBTENIDOS

En este capítulo se presentan los resultados obtenidos durante las pruebas realizadas con el simulador de conducción. Se analizarán los datos recopilados a través del test de resistencia simple y el test de resistencia a la monotonía, detallando los hallazgos obtenidos.

Para el test de reacción simple se contará con la participación de seis voluntarios, cada uno de los cuales realizará los distintos tests para evaluar sus habilidades de reacción. El objetivo principal es determinar el nivel de esta habilidad en cada colaborador.

Se llevarán a cabo dos sesiones de pruebas para cada participante: una en la mañana a las 9:00 am y otra en la tarde a las 6:00 pm. Esto permitirá analizar si existen diferencias significativas en los resultados debido a la fatiga o el cansancio del usuario a lo largo del día.

Los resultados de cada participante serán verificados y tabulados utilizando Microsoft Excel. Cada test realizado será sometido a una evaluación para de esta manera establecer el nivel de logro de cada participante en el mismo. Cada test constará de 10 intentos.

Posteriormente, se calculará un promedio de los resultados obtenidos en cada sesión de pruebas para cada participante. Este promedio proporcionará una evaluación final que indicará si el participante pasa o no la prueba en términos de habilidades de reacción.

Este enfoque metodológico garantiza un análisis exhaustivo y riguroso de las habilidades evaluadas, considerando tanto la variabilidad individual como los efectos potenciales de la fatiga en los resultados de las pruebas.

2.1. Análisis e interpretación de resultados del test de reacción simple

Con las 6 personas voluntarias las cuales van a realizar el test de resistencia simple son:

Tabla 2.1: Resultados del Participante 1 Steven Morillo 24 años Hora 9am

Prueba	Tiempo	Velocidad	Resultado
1	0,21	30	APROBADO
2	0,36	20	APROBADO
3	0,38	30	APROBADO
4	0,37	15	APROBADO
5	0,41	20	APROBADO
6	0,29	30	APROBADO
7	0,42	15	APROBADO
8	0,74	20	NO APROBADO
9	0,42	15	APROBADO
10	0,28	30	APROBADO

En la tabla se evidencia que el participante 1 ha cometido errores en solo el 10% de las pruebas, logrando un promedio general de aprobación del 90%. Cada prueba presentó distintos tipos de obstáculos diseñados para medir el tiempo de reacción del participante. Es crucial mencionar que las pruebas fueron realizadas por la mañana, cuando el participante se encontraba en un estado de descanso y antes de haber empleado su energía en actividades laborales.

Tabla 2.2: Resultado del participante 1 Steven Morillo 24 años, 6pm

Prueba	Tiempo	Velocidad	Resultado
1	0,29	30	APROBADO
2	0,28	20	APROBADO
3	0,63	30	NO APROBADO
4	0,48	15	NO APROBADO
5	0,31	20	APROBADO
6	0,43	30	APROBADO
7	0,46	15	NO APROBADO
8	0,35	20	APROBADO
9	0,35	15	APROBADO
10	0,50	30	APROBADO

En la sesión de la tarde, el participante 1 demostró un índice de aprobación del 70%, con un 30% de pruebas no superadas. A pesar de cumplir con el tiempo de reacción requerido en

promedio, se observó un incremento en el número de fallos respecto a los resultados de la mañana. Este fenómeno sugiere que el desgaste asociado al día laboral podría influir negativamente en las habilidades de reacción del individuo durante las pruebas.

Tabla 2.3: Resultado del participante 2 Christian Sangucho 23 años Hora 9am

Prueba	Tiempo	Velocidad	Resultado
1	0,25	30	APROBADO
2	0,41	20	APROBADO
3	0,34	30	APROBADO
4	0,43	15	APROBADO
5	0,58	20	NO APROBADO
6	0,32	30	APROBADO
7	0,41	15	APROBADO
8	0,39	20	APROBADO
9	0,43	15	APROBADO
10	0,33	30	APROBADO

En la mañana, el participante 2 demostró un alto índice de aprobación del 90%, con solo un 10% de pruebas no aprobadas. Estos resultados reflejan su estado óptimo de alerta y concentración, beneficiado por un descanso adecuado y la frescura de sus estímulos cognitivos.

Tabla 2.4: Resultado del participante 2 Christian Sangucho 23 años Hora 6pm

Prueba	Tiempo	Velocidad	Resultado
1	0,51	30	NO APROBADO
2	0,28	20	APROBADO
3	0,30	30	APROBADO
4	0,48	15	NO APROBADO
5	0,42	20	APROBADO
6	0,23	30	APROBADO
7	1,17	15	NO APROBADO
8	0,39	20	APROBADO
9	0,44	15	NO APROBADO
10	0,33	30	APROBADO

En la tarde, el participante 2 mostró un resultado donde el 40% de las pruebas no cumplió con el tiempo establecido, mientras que el 60% sí lo hizo. Estos datos muestran una notable

diferencia respecto a la sesión matutina, reflejando una mayor incidencia de errores debido al cansancio después de una jornada laboral de 8 horas.

Tabla 2.5: Resultado del participante 3 Sofia Sangucho 17 años. Hora: 9am

Prueba	Tiempo	Velocidad	Resultado
1	0,31	30	APROBADO
2	0,48	20	NO APROBADO
3	0,45	30	NO APROBADO
4	0,40	15	APROBADO
5	0,36	20	APROBADO
6	0,31	30	APROBADO
7	0,36	15	APROBADO
8	0,39	20	APROBADO
9	0,42	15	APROBADO
10	0,46	30	NO APROBADO

En la mañana, el participante 3 mostró un índice de aprobación del 70%, con un 30% de pruebas no cumplidas dentro del tiempo establecido. Aunque logró aprobar, los resultados no alcanzaron el nivel esperado, posiblemente debido a su falta de familiaridad con el programa, agravado por su edad.

Tabla 2.6: Resultado del participante 3 Sofia Sangucho 17 años. Hora: 18pm

Prueba	Tiempo	Velocidad	Resultado
1	0,34	30	APROBADO
2	0,30	20	APROBADO
3	0,31	30	APROBADO
4	0,29	15	APROBADO
5	0,38	20	APROBADO
6	0,41	30	APROBADO
7	0,40	15	APROBADO
8	0,36	20	APROBADO
9	0,38	15	APROBADO
10	0,45	30	NO APROBADO

En la tarde, el participante 3 mostró una notable mejora, con un índice de aprobación del 90% y solo un 10% de pruebas no cumplidas dentro del tiempo establecido. A diferencia de la sesión matutina, donde su falta de familiaridad con el simulador afectó su desempeño, en

la tarde demostró una drástica mejoría en su efectividad, indicando que el cansancio no afectó negativamente su rendimiento.

Tabla 2.7: Resultado del participante 4 Félix Morillo 62 años. Hora: 9am

Prueba	Tiempo	Velocidad	Resultado
1	0,35	30	APROBADO
2	1,50	20	NO APROBADO
3	0,46	30	NO APROBADO
4	0,47	15	NO APROBADO
5	0,49	20	NO APROBADO
6	0,31	30	APROBADO
7	0,43	15	APROBADO
8	0,27	20	APROBADO
9	0,37	15	APROBADO
10	0,39	30	APROBADO

En la mañana, el participante 4 no logró cumplir con el tiempo de reacción en un 40% de las pruebas, mientras que en el 60% restante sí lo hizo. Esto indica que no cumplió con los parámetros establecidos para aprobar el test, posiblemente debido a limitaciones relacionadas con su edad y la capacidad de reacción reducida.

Tabla 2.8: Resultado del participante 4 Félix Morillo 62 años. Hora: 18pm

Prueba	Tiempo	Velocidad	Resultado
1	0,31	30	APROBADO
2	0,31	20	APROBADO
3	0,37	30	APROBADO
4	0,50	15	NO APROBADO
5	0,60	20	NO APROBADO
6	0,32	30	APROBADO
7	0,31	15	APROBADO
8	0,47	20	NO APROBADO
9	2,90	15	NO APROBADO
10	0,45	30	NO APROBADO

En la tarde, el participante 4 mostró un resultado dividido: el 50% de las pruebas fueron aprobadas, mientras que el otro 50% no cumplió con los tiempos establecidos. Este cambio

podría atribuirse a una mayor fatiga experimentada por el participante en la sesión vespertina.

Tabla 2.9: Resultado del participante 5 María Ordoñez 34 años. Hora: 9am.

Prueba	Tiempo	Velocidad	Resultado
1	0,50	30	NO APROBADO
2	0,38	20	APROBADO
3	0,36	30	APROBADO
4	0,43	15	APROBADO
5	0,27	20	APROBADO
6	0,45	30	NO APROBADO
7	1,06	15	NO APROBADO
8	0,46	20	NO APROBADO
9	0,38	15	APROBADO
10	0,38	30	APROBADO

En la mañana, el participante 5 mostró un rendimiento donde el 40% de las pruebas no cumplió con los tiempos establecidos, mientras que el 60% sí lo logró. Esto se atribuye a un problema en el pie reportado por la participante, posiblemente afectando su capacidad de reacción en algunas pruebas.

Tabla 2.10: Resultado del participante 5 María Ordoñez 34 años. Hora: 18pm.

Prueba	Tiempo	Velocidad	Resultado
1	0,48	30	NO APROBADO
2	0,60	20	NO APROBADO
3	0,59	30	APROBADO
4	0,39	15	APROBADO
5	0,45	20	NO APROBADO
6	0,33	30	APROBADO
7	0,40	15	APROBADO
8	0,41	20	APROBADO
9	0,37	15	APROBADO
10	1,74	30	NO APROBADO

En la tarde, el participante 5 mostró un rendimiento donde aprobó el 60% de las pruebas, mientras que en el 40% restante no logró cumplir los tiempos establecidos. Esto se debe a

que, aunque se le dieron 10 oportunidades en cada prueba, al final se obtiene un promedio que refleja su desempeño general.

Tabla 2.11: Resultado del participante 6 Patricio Oñate 29 años. Hora:9am

Prueba	Tiempo	Velocidad	Resultado
1	0,33	30	APROBADO
2	0,39	20	APROBADO
3	0,30	30	APROBADO
4	0,46	15	NO APROBADO
5	0,29	20	APROBADO
6	0,38	30	APROBADO
7	0,38	15	APROBADO
8	0,29	20	APROBADO
9	0,46	15	NO APROBADO
10	0,29	30	APROBADO

En la mañana, el participante 6 mostró un rendimiento donde el 20% de las pruebas no cumplió con los tiempos establecidos, mientras que el 80% sí lo hizo. Estos resultados indican que el participante estaba completamente descansado y pudo rendir bien durante las pruebas realizadas en ese horario.

Tabla 2.12: Resultado del participante 6 Patricio Oñate 29 años. Hora:18pm

Prueba	Tiempo	Velocidad	Resultado
1	0,43	30	APROBADO
2	0,29	20	APROBADO
3	0,42	30	APROBADO
4	0,40	15	APROBADO
5	0,35	20	APROBADO
6	0,34	30	APROBADO
7	0,42	15	APROBADO
8	0,31	20	APROBADO
9	0,33	15	APROBADO
10	0,31	30	APROBADO

El último participante cumplió con todos los tiempos establecidos en cada prueba y aprobó exitosamente. Este rendimiento perfecto sugiere que el participante adquirió un conocimiento completo del funcionamiento del simulador, lo cual contribuyó a obtener una puntuación impecable en todas las pruebas realizadas.

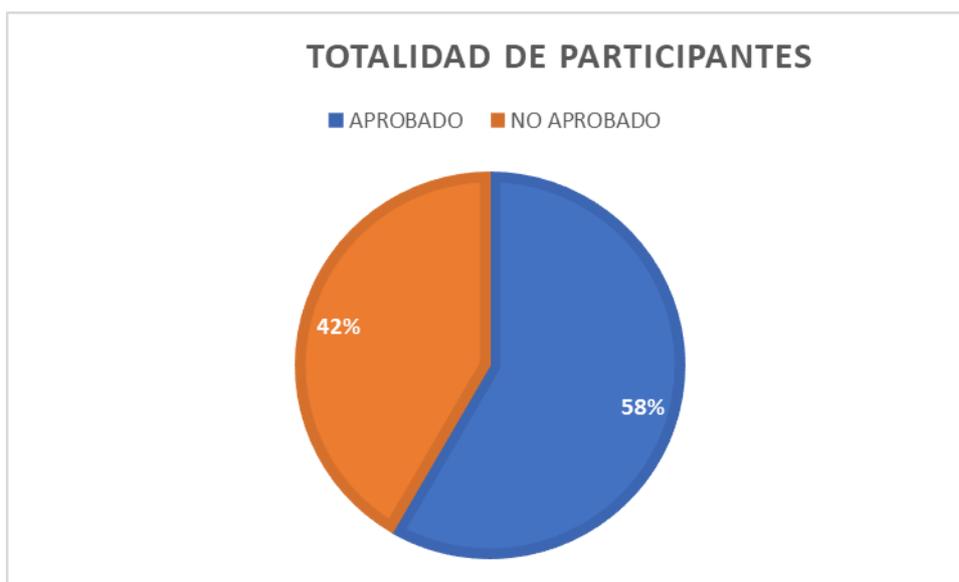
Tabla 2.13: Resultados del test de resistencia simple

INTENTOS	1		2		3		4		5		6		7		8		9		10		RESULTADOS	
PARTICIPANTE	T	V	T	V	T	V	T	V	T	V	T	V	T	V	T	V	T	V	T	V		
1 MAÑANA	0,21	30	0,36	20	0,38	30	0,37	15	0,41	20	0,29	30	0,42	15	0,74	20	0,42	15	0,28	30	0,41	APROBADO
1 TARDE	0,29	30	0,28	20	0,63	30	0,48	15	0,31	20	0,43	30	0,46	15	0,35	20	0,35	15	0,35	30	0,42	APROBADO
2 MAÑANA	0,25	30	0,41	20	0,34	30	0,43	15	0,58	20	0,32	30	0,41	15	0,39	20	0,43	15	0,33	30	0,42	APROBADO
2 TARDE	0,51	30	0,28	20	0,30	30	0,48	15	0,42	20	0,23	30	1,17	15	0,39	20	0,44	15	0,33	30	0,46	NO APROBADO
3 MAÑANA	0,31	30	0,48	20	0,45	30	0,40	15	0,36	20	0,31	30	0,36	15	0,39	20	0,42	15	0,46	30	0,39	APROBADO
3 TARDE	0,34	30	0,30	20	0,31	30	0,29	15	0,38	20	0,41	30	0,40	15	0,36	20	0,38	15	0,45	30	0,36	APROBADO
4 MAÑANA	0,35	30	1,50	20	0,46	30	0,47	15	0,49	20	0,31	30	0,43	15	0,27	20	0,37	15	0,39	30	0,46	NO APROBADO
4 TARDE	0,31	30	0,31	20	0,37	30	0,50	15	0,60	20	0,32	30	0,31	15	0,47	20	2,09	15	0,45	30	0,55	NO APROBADO
5 MAÑANA	0,50	30	0,38	20	0,37	30	0,43	15	0,27	20	0,45	30	1,06	15	0,46	20	0,38	15	0,38	30	0,47	NO APROBADO
5 TARDE	0,38	30	0,29	20	0,59	30	0,39	15	0,45	20	0,33	30	0,40	15	0,41	20	0,37	15	1,74	30	0,54	NO APROBADO
6 MAÑANA	0,33	30	0,39	20	0,30	30	0,46	15	0,29	20	0,38	30	0,38	15	0,29	20	0,46	15	0,49	30	0,40	APROBADO
6 TARDE	0,43	30	0,29	20	0,42	30	0,40	15	0,35	20	0,34	30	0,42	15	0,31	20	0,33	15	0,31	30	0,38	APROBADO

En la tabla 2.13 se presentan los resultados del test de resistencia simple para seis participantes, evaluados tanto en la mañana como en la tarde. Cada participante realizó diez intentos, registrándose los tiempos de reacción (T) y velocidades (V) para cada uno. Los resultados indican que cuatro de los seis participantes aprobaron todas las pruebas en ambos horarios, mientras que dos participantes no aprobaron en la tarde.

Un análisis detallado de los promedios y desviaciones estándar sugiere una variabilidad en el rendimiento a lo largo del día. Específicamente, los resultados de la tarde se vieron negativamente afectados, debido al cansancio acumulado por las actividades diarias, como el trabajo y otras obligaciones. Esto sugiere que el horario influye significativamente en los tiempos de reacción y velocidades de respuesta.

Figura 2.1: Totalidad de participantes



En la figura 2.1 se evidencia que el 42% de los participantes no aprobaron la prueba de resistencia simple, mientras que el 58% de ellos la superaron exitosamente. Esta evaluación se basa en un límite de 0,43 segundos por prueba, tal como se detalla en la tabla 2.13. En dicha tabla se presentan los resultados individuales, incluyendo los tiempos y las velocidades alcanzadas por cada participante.

2.2. Análisis e interpretación de resultados del test de resistencia a la monotonía

En esta sección se presentan los resultados del test de resistencia a la monotonía, el cual involucra a 12 participantes. Este test se llevó a cabo utilizando dos componentes esenciales:

la pedalera y la botonera. Antes de iniciar las pruebas, se realizó una inducción detallada para familiarizar a los participantes con el procedimiento y las instrucciones específicas del test. Es crucial que los participantes memoricen estas indicaciones iniciales, ya que su cumplimiento es fundamental para la verificación precisa de cada prueba.

Tabla 2.14: Resultados del participante 1 Erick Lascano 23 años,

Prueba	Tiempo	Resultado
0	4	APROBADO
1	2	APROBADO
2	3	APROBADO
3	2	APROBADO
4	3	APROBADO
5	4	APROBADO
6	2	APROBADO
7	4	APROBADO
8	1	APROBADO

El participante 1 cumplió con los tiempos establecidos en el 100% de las pruebas del test de resistencia a la monotonía. Demostró un buen nivel de memorización de las indicaciones proporcionadas antes de la aplicación del test.

Tabla 2.15: Resultados del participante 2 Fernanda Merino 30 años

Prueba	Tiempo	Resultado
0	3	APROBADO
1	0	NO APROBADO
2	2	APROBADO
3	2	APROBADO
4	1	NO APROBADO
5	2	APROBADO
6	3	APROBADO
7	3	NO APROBADO
8	2	APROBADO

El participante 2 aprobó 6 de las pruebas del test de resistencia a la monotonía, cumpliendo adecuadamente con los tiempos establecidos y mostrando buena concentración.

Sin embargo, en 3 pruebas no logró aprobar debido a que no cumplió con los tiempos requeridos y no prestó la atención necesaria a las instrucciones.

Tabla 2.16: Resultados del participante 3 Miguel Obando 74 años.

Prueba	Tiempo	Resultado
0	3	NO APROBADO
1	3	NO APROBADO
2	2	APROBADO
3	2	APROBADO
4	1	NO APROBADO
5	4	APROBADO
6	2	APROBADO
7	2	APROBADO
8	2	APROBADO

El participante 3 no cumplió con el tiempo en 3 pruebas y mostró falta de atención a las instrucciones. Sin embargo, aprobó 6 pruebas. Con base en estos resultados y su promedio general, el participante logró aprobar el test de resistencia a la monotonía.

Tabla 2.17: Resultados del participante 4 Alisson Robalino 18 años.

Prueba	Tiempo	Resultado
0	2	NO APROBADO
1	2	APROBADO
2	5	APROBADO
3	2	APROBADO
4	0	NO APROBADO
5	5	APROBADO
6	2	APROBADO
7	3	APROBADO
8	3	APROBADO

Como se evidencia en la tabla anterior del participante 4 no aprueba la 0 y 4, mientras que las demás pruebas la cumplen con los tiempos establecidos y con el promedio realizado pasa el test de resistencia a la monotonía.

Tabla 2.18: Resultados del participante 5 Neiser Bravo 17 años.

Prueba	Tiempo	Resultado
0	3	APROBADO
1	4	APROBADO
2	3	APROBADO
3	4	APROBADO
4	3	NO APROBADO
5	2	APROBADO
6	2	NO APROBADO
7	1	NO APROBADO
8	2	APROBADO

El participante 5 se equivoca en las pruebas 4, 6 y 7 no aprueba no cumple con el tiempo establecido de reacción, y las otras si aprueba con los tiempos establecidos de acuerdo con lo estipulado y con las indicaciones de la cartilla de apoyo.

Tabla 2.19: Resultados del participante 6 Ana Salinas 20 años.

Prueba	Tiempo	Resultado
0	3	APROBADO
1	2	APROBADO
2	3	APROBADO
3	2	APROBADO
4	2	APROBADO
5	0	NO APROBADO
6	1	APROBADO
7	3	NO APROBADO
8	2	APROBADO

En las pruebas 5 y 7, el participante 6 no logró aprobar debido a falta de concentración en las instrucciones y por no cumplir con los tiempos establecidos. Sin embargo, en las demás pruebas demostró un desempeño adecuado, lo cual, junto con su promedio general, le permitió aprobar el test de resistencia a la monotonía.

Tabla 2.20: Resultados del participante 7 Kevin Acurio 23 años.

Prueba	Tiempo	Resultado
0	2	NO APROBADO
1	3	APROBADO
2	1	APROBADO
3	4	APROBADO
4	6	NO APROBADO
5	2	APROBADO
6	3	APROBADO
7	2	APROBADO
8	2	APROBADO

El participante 7 cometió errores en los ítems 0 y 4 debido a la falta de concentración en las instrucciones proporcionadas por el instructor del test. Sin embargo, aprobó los demás ítems al cumplir con los tiempos establecidos.

Tabla 2.21: Resultados del participante 8 Daniel Zambrano 21 años.

Prueba	Tiempo	Resultado
0	3	APROBADO
1	2	APROBADO
2	3	APROBADO
3	0	NO APROBADO
4	3	APROBADO
5	3	APROBADO
6	4	APROBADO
7	5	APROBADO
8	1	NO APROBADO

Por otro lado, el participante 8 no logró aprobar los tests 3 y 8, lo que indica que no cumplió con los tiempos específicos establecidos para cada prueba de reacción inmediata. Sin embargo, las respuestas a los demás tests mostraron una estructura adecuada.

Tabla 2.22: Resultados del participante 9 María Ordoñez 34 años

Prueba	Tiempo	Resultado
0	7	NO APROBADO
1	2	APROBADO
2	2	APROBADO
3	2	APROBADO
4	3	APROBADO
5	2	APROBADO
6	2	APROBADO
7	1	NO APROBADO
8	4	APROBADO

El participante 9 no aprobó dos pruebas debido a la falta de atención, sin embargo, logró aprobar las otras 7 pruebas. Con base en el promedio de todos los tiempos y resultados obtenidos, el participante aprobó el test de resistencia a la monotonía.

Tabla 2.23: Resultados del participante 10 Junior Bermúdez 28 años.

Prueba	Tiempo	Resultado
0	3	APROBADO
1	3	APROBADO
2	1	APROBADO
3	3	APROBADO
4	3	APROBADO
5	2	APROBADO
6	1	NO APROBADO
7	3	NO APROBADO
8	1	APROBADO

En la prueba 6 y 7 el participante 10 no aprueba por el tiempo de reacción, mientras que las demás 7 pruebas son aprobadas de acuerdo con el promedio del test.

Tabla 2.24: Resultados del participante 11 Patricio Oñate 29 años.

Prueba	Tiempo	Resultado
0	2	APROBADO
1	4	APROBADO
2	2	APROBADO
3	3	NO APROBADO
4	0	NO APROBADO
5	4	APROBADO
6	2	APROBADO
7	2	APROBADO
8	4	APROBADO

Por otra parte, el participante 11 no aprueba las pruebas 3 y 4 por falta de tiempo, y las demás pruebas si lo hace de acuerdo con el tiempo de reacción de cada una, por esa razón mediante su promedio correspondiente aprueba el test.

Tabla 2.25: Resultados del participante 12 Nazareno Tipán 20 años.

Prueba	Tiempo	Resultado
0	2	APROBADO
1	4	APROBADO
2	4	APROBADO
3	7	NO APROBADO
4	2	NO APROBADO
5	5	APROBADO
6	1	APROBADO
7	5	APROBADO
8	2	APROBADO

Se puede verificar en la tabla que el participante 12 se equivoca de igual manera en dos pruebas la 3 y 4 las cuales no son aprobadas mientras que en las otras si, y con el promedio correspondiente si pasa el test.

Tabla 2.26: Resultados de todos los participantes

PARTICIPANTE	1	2	3	4	5	6	7	8	9	10	11	12
PRUEBA	T	T	T	T	T	T	T	T	T	T	T	T
1	4	3	3	2	3	3	2	3	7	3	2	2
2	2	0	3	2	4	2	3	2	2	3	4	4
3	3	2	2	5	3	3	1	3	2	1	2	4
4	2	2	2	2	4	2	4	0	2	3	3	7
5	3	1	1	0	3	2	6	3	3	3	0	2
6	4	2	4	5	2	0	2	3	2	2	4	5
7	2	3	2	2	2	1	3	4	2	1	2	1
8	4	3	2	3	1	3	2	5	1	3	2	5
PROMEDIO	3	2	2.37	2.62	2,75	2	2.87	2.87	2.62	2.37	2.37	3.75

La tabla 2.26 presenta los resultados de los 12 participantes que realizaron el test de resistencia a la monotonía, distribuido en 8 pruebas diferentes. Los puntajes obtenidos reflejan la capacidad de los participantes para mantener la atención y concentración a lo largo de las pruebas. Se observa una variabilidad en los promedios de los puntajes, que oscilan entre 2 y 3.75, indicando diferencias en el rendimiento de los participantes.

El análisis sugiere que la capacidad de concentración y memorización fueron fundamentales para que los participantes aprobaran el test. La resistencia a la monotonía requiere que los individuos mantengan su atención y recuerden patrones específicos durante periodos prolongados. La falta de atención y concentración resultó en un rendimiento inferior, como se refleja en los puntajes más bajos.

Estos resultados subrayan la importancia de desarrollar estrategias para mantener la atención en tareas monótonas, especialmente en áreas críticas como la conducción, donde la capacidad para resistir la monotonía puede tener un impacto significativo en la seguridad.

Figura 2.2: Resultados de todos los participantes



Se puede evidenciar que durante la prueba algunos no aprueban en ciertos numerales, por no tener concentración y no poner atención a las instrucciones respectivas y esto no permitió que reacciones en el tiempo adecuado, esto quiere decir que de acuerdo con el procedimiento el 100% de los participantes aprobaron el test de resistencia a la monotonía.

CONCLUSIONES

- Los objetivos propuestos en el proyecto fueron cumplidos satisfactoriamente. El simulador desarrollado no solo demostró ser efectivo en la evaluación de habilidades críticas para la conducción, sino que también cumplió con los requisitos legales y normativos establecidos, asegurando su aplicabilidad y relevancia para la formación de futuros estudiantes.
- Los resultados del test de reacción simple mostraron que los participantes experimentaron una variabilidad significativa en sus tiempos de reacción y velocidades según el horario del día. Específicamente, se observó que el rendimiento en la tarde fue inferior debido al cansancio acumulado por las actividades diarias. Esto resalta la importancia de considerar el momento del día al evaluar habilidades críticas como la reacción en situaciones de conducción.
- El test de resistencia a la monotonía reveló que los participantes presentaron variabilidad en su capacidad para mantener la atención y concentración durante las pruebas. Los puntajes más bajos estuvieron asociados con lapsos de atención reducidos y dificultad para recordar patrones específicos durante periodos prolongados. Esto subraya la necesidad de desarrollar estrategias efectivas para mejorar la resistencia a la monotonía, especialmente en contextos críticos como la conducción.
- El simulador proporciono un entorno seguro para que los participantes puedan practicar y mejorar sus habilidades de conducción, mediante la simulación de escenarios realistas de tráfico, preparando a los conductores para enfrentar situaciones reales en la carretera de manera más segura y confiada.

RECOMENDACIONES

- Es recomendable realizar más pruebas y validar el simulador con un mayor número de estudiantes y bajo diferentes condiciones. Esto permitirá consolidar los hallazgos y asegurar la efectividad del simulador en diversas situaciones.
- Evaluar la posibilidad de ampliar las funcionalidades del simulador para incluir más escenarios de conducción y evaluar otras habilidades relevantes para los conductores, como la gestión del estrés y la toma de decisiones rápidas. Como pruebas auditivas y pruebas de visión.
- Considerar la implementación de horarios de prueba flexibles que permitan a los participantes realizar las evaluaciones en momentos del día en los que estén más alertas y menos afectados por el cansancio. Esto podría mejorar la precisión de las evaluaciones y proporcionar datos más representativos del rendimiento real de los conductores.
- Realizar periódicamente pruebas para poder identificar patrones significativos y tendencias en el rendimiento de los conductores. Este análisis puede revelar información adicional sobre factores críticos que afecten la seguridad.
- El simulador puede ser utilizado en la investigación en seguridad vial, ya que cuenta con herramientas como la recolección de datos del rendimiento de los conductores, la simulación de escenarios variados y la capacidad de reproducir condiciones adversas de conducción de manera controlada y segura.

REFERENCIAS BIBLIOGRÁFICAS.

- BIBLIOGRAPHY Agencia Nacional de Tránsito (ANT). (26 de Enero de 2023). *Gob.Ec.*
Obtenido de <https://www.gob.ec/ant/tramites/homologacion-simuladores-conduccion-equipos-psicosensometricos-personas-juridicas-importados>
- Agencia Nacional de tránsito A.N.T. (2011, 29 de Marzo). *Ley Orgánica de Transporte Terrestre, tránsito y Seguridad Vial*. El Registro Oficial.
- Aneta. (18 de Marzo de 2021). *Aneta.org.ec*. Obtenido de <https://www.aneta.org.ec/evaluacion-psicosensometrica/>
- Betaksur. (27 de Noviembre de 2023). *ANT Ecuador*. Recuperado el 15 de Junio de 2023, de <https://ant.com.ec/blog/ver/como-sacar-el-examen-psicosensometrico>
- Biartic. (2021). *studocu*. Obtenido de <https://www.studocu.com/pe/document/universidad-cesar-vallejo/psicopatologia/psicosensometrico/7178877>
- BIARTIC. (s.f.). *Biartic E value*. Recuperado el 14 de Mayo de 2023, de <https://www.biartic.com/Documentacion/01-Documentacion,%20manuales%20y%20caracter%20C3%ADsticas/03-Baremos%20generales.pdf>
- Bort, H. M. (s.f.). *Journals*. Recuperado el 24 de Mayo de 2023, de La evaluación psicológica por medio del ordenador:
<https://journals.copmadrid.org/jwop/files/1987/vol1/arti3.htm>
- Cárdenas, M. (22 de Octubre de 2021). *Docplayer*. Obtenido de <https://docplayer.es/210909789-Psicosensometrico-precision-tecnologia-seguridad-evaluacion-psicométrica-evaluacion-sensométrica.html>
- Federación del Transporte. (02 de Agosto de 2018). *FROET*. Obtenido de <https://www.froet.es/simuladorfroet.pdf>
- Flores, M. Á. (Enero de 2018). *Docplayer*. Obtenido de <https://docplayer.es/86090266-Gabinete-psicosensometrico.html>
- Grupo GMA. (s.f.). *GrupoMedico de Antioquia*. Recuperado el 30 de Mayo de 2023, de <https://www.grupomedicodeantioquia.com/blog/examen-pesv-quien-se-le-realiza/>
- Grupo NW. (11 de Julio de 2018). *Sanitco*. Obtenido de https://www.sanitco.com/blog&pagina_actual=5
- NORCLAN. (2000). *Psicotecnicos*. Recuperado el 04 de Junio de 2023, de <https://psicotecnicos.cl/equipo-psicotecnico-conductores-brevetes/>

- SCIELO. (2008). Valoración de la aptitud psicomotora y la inteligencia requerida para conducir en los Centros de Reconocimiento de Conductores. En M. OZCOIDI, M. SANZ, C. CIVERA, & MONTORO, *Intervención Psicosocial* (Vol. XVII, págs. 23-32). Madrid: Cuesta de San Vicente. Obtenido de <https://scielo.isciii.es/pdf/inter/v17n1/v17n1a03.pdf>
- Analuisa, J. (2019). *Gestión en Seguridad Vial*.
- Andrade, J., & Gonzales, I. (2020). *Conducción Aplicado a Conductores De Vehículos*.
- ANT. (2021). Minería de datos para determinar los factores más influyentes en la ocurrencia de siniestros de tránsito en Ecuador en el año 2020. *CEDAMAZ*, 11(2), 124–132. <https://doi.org/10.54753/cedamaz.v11i2.1181>
- Biartic. (2022a). Test de reacción simple. *BIARTIC*.
- Biartic. (2022b). Test de resistencia a la monotonía. *BIARTIC*.
- Biartic. (2022c). Test de resistencia a la monotonía. *BIARTIC*.
- Bima, D. (2023). *Biartic.cdr*.
- Diario Primicias. (2023a). *Los accidentes de tránsito son cada vez más violentos en Ecuador*.
- Diario Primicias. (2023b). *Los accidentes de tránsito son cada vez más violentos en Ecuador*.
- García, J., Velázquez, J., & Gonzáles, A. (2019). *Educación vial y habilidades cognitivas*.
- Herrero, J. (2020). *Desarrollo e implementación de algoritmos de conducción autónoma en simuladores*.
- Jerez, C., Torres, A., & Beltrán, M. (2020). Criterios para la Evaluación de Fatiga en Conductores de Transporte. *Paper Knowledge . Toward a Media History of Documents*.
- Microsoft. (2022). Guía de escritorio (.NET para Windows Forms). *Microsoft*.
- Microsoft. (2023a). Visual Studio 2022. *Microsoft*.
- Microsoft. (2023b). Visual Studio 2022. *Microsoft*.
- Nhtsa. (2024). *Crash•Stats: Early Estimates of Motor Vehicle Traffic Fatalities and Fatality Rate by Sub-Categories in 2023 1 Early Estimates of Motor Vehicle Traffic Fatalities and Fatality Rate by Sub-Categories in 2023 Introduction and Summary*.
- Osorio, R., & Vera, R. (2021). *Desarrollo de un simulador para el aprendizaje de conducción vehicular utilizando un motor de videojuego*.
- Patricio Castillo Villacrés Eugenio Javier Escobar Gonzáles, H., & del Carmen Viteri Naranjo, B. (2020). *Estudios del Desarrollo Social: Cuba y América Latina Validación*

mediante criterio de expertos de proyecto de reforma al artículo 238 del Reglamento General de la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial de Ecuador Validation through Expert Judgment of a Project to Reform Article 238 of the General Regulations of the Organic Law of Land Transport, Transit and Road Safety of Ecuador. In *Estudios del Desarrollo Social: Cuba y América Latina RPNS* (Vol. 2346). www.revflacso.uh.cu

RAUCH. (2019). *Manual de conducción*. 1–62.

Seguridad vial. (2012). *Reglamento a ley de transporte terrestre*. 1–91.

Servicio Integrado de Seguridad. (2023). *En 2023, al 9-1-1 se han reportado 32.687 accidentes de tránsito*.

Vera, L. (2019). *La aptitud perceptivo motora en la conducción*.

ANEXOS

Presupuesto

Los valores detallados son los utilizados en el presente proyecto para el diseño y elaboración del simulador de conducción.

Figura 2.3: Presupuesto

Tipo	Descripción	Valor Estimado (\$)
Equipos	Tarjetas Electrónicas, Equipos de Medición. Software Especializado.	200
Materiales	Elementos Electrónicos, Elementos mecánicos, Electromecánicos, Químicos,	400
Herramientas	Pinzas, alicates, cautín, etc.	45
Capacitación	Costo por horas por cursos.	150
Asesoría Externa	Costo por horas de tutor externo.	0
Movilización	Gastos por traslado al sitio del proyecto	40
VALOR TOTAL ESTIMADO		835

Los valores anteriormente expuestos son los que se utilizaron para la realización y ejecución del presente proyecto; el simulador servirá para los estudiantes de Ingeniería Automotriz para poder revisar su nivel de reacción ante un caso durante la conducción, ya que la reacción simple permite visualizar el tiempo de reacción que se tiene con los pedales y la de resistencia a la monotonía lo que permite conocer si en viajes largos el conductor se encuentra concentrado y atento a la situación de la carretera.