

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

Desarrollo de una herramienta informática en MATLAB para la estandarización de la toma de datos en estudios de vida pública.

Tesis para optar por el Título de Ingeniero Civil, que presenta el bachiller:

José Antonio Mallma León

ASESOR: Félix Israel Cabrera Vega

Lima, marzo de 2017



RESUMEN

El interés por el estudio de los peatones ha ido aumentando a lo largo de los años debido a su importancia en la movilidad y en el diseño del espacio público. El proceso de diseño se inicia con el estudio de vida pública que permite recolectar información sobre los patrones de comportamiento de las personas. Se registran flujos y velocidades peatonales, patrones de movimiento, lugares de estancia, etc., pero a la fecha no existe un software que permita sistematizar la información recolectada y organizarla de manera estándar.

Por ello, el presente trabajo tiene como objetivo principal el desarrollo e implementación de un software, que permita el registro de los datos recolectados en estudios de vida pública. Dicho programa cuenta con una interfaz gráfica de usuario para facilitar su uso. El programa fue desarrollado en el entorno del lenguaje de programación MATLAB. Se implementaron seis herramientas de los estudios de vida pública: Mapping para mapear las actividades de las personas, Tracing para identificar las rutas o líneas de deseo, Tracking para registrar el seguimiento de los peatones, Counting para registrar los flujos peatonales, Speed Test para evaluar las velocidades de las personas, y Diary que permite registrar información genérica que no pertenece a las demás herramientas.

Para tal fin, se adoptó el proceso típico de la ingeniería de software que consta de etapas parcialmente ordenadas. En primer lugar, se traducen las necesidades del usuario en requerimientos de software, luego estos son transformados en diseño, dicho diseño es implementado en código. Finalmente, el código es probado y documentado.

Con base en las pruebas realizadas, se concluye que el programa cumplió con las expectativas y objetivos planteados; sin embargo, es importante resaltar que se trata de una primera versión y tiene bastantes limitaciones. Se espera que en un futuro se pueda mejorar y ampliar las funcionalidades con las que se cuenta de acuerdo a los nuevos sujetos de estudios que aparezcan en la movilidad.

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO CIVIL

Título : "Desarrollo de una herramienta informática en MATLAB para la estandarización de la toma de datos en estudios de vida pública"

Área : Movilidad y Transporte

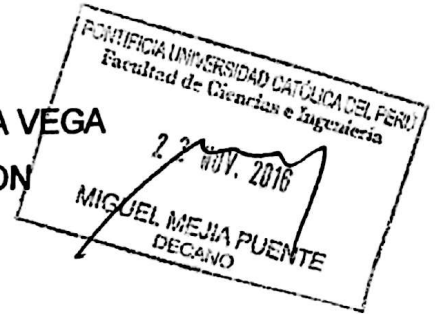
Asesor : ING. FELIX ISRAEL CABRERA VEGA

Alumno : JOSE ANTONIO MALLMA LEON

Código : 2011.1300.412

Tema N° : # 60

Fecha : Lima, 21 de noviembre del 2016



INTRODUCCIÓN

El interés por el estudio de los peatones ha ido aumentando a lo largo de los años debido a su importancia en la movilidad y en el diseño del espacio público. Un adecuado proceso de diseño se inicia con el estudio de la vida pública que permite recolectar información sobre los patrones de comportamiento de las personas en la ciudad. Se registran los flujos peatonales, velocidades, líneas de movimiento, lugares de estancia, seguimiento, etc., pero hasta el momento no existe un software que permita sistematizar la información recolectada y que además permita almacenar la información de diferentes proyectos.

Por ello, en esta tesis, se elabora una aplicación con interfaz gráfica, en código MATLAB, que permite cumplir con los requerimientos mencionados anteriormente. Se resalta que esta aplicación es una primera versión y que se espera que en un futuro pueda ser mejorada de acuerdo a los nuevos sujetos de estudios que aparezcan en la movilidad.

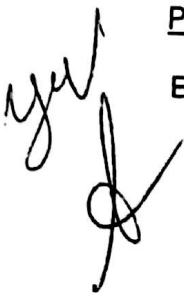
OBJETIVO

Desarrollar una herramienta informática, basada en MATLAB, y con una interfaz gráfica de usuario, que permita procesar los datos recolectados mediante las herramientas de los estudios de vida pública.

PROGRAMA DE TRABAJO

El desarrollo de la tesis considerará los siguientes temas:

- a) Objetivos
- b) Revisión de la literatura: del tránsito a la movilidad, espacio público, vida pública, estudios de vida pública
- c) Metodología
- d) Análisis de los requerimientos
- e) Diseño del programa
- f) Implementación
- g) Periodo de prueba.
- h) Conclusiones y recomendaciones



METODOLOGÍA

Se adopta el proceso típico de la ingeniería de software que consta de etapas parcialmente ordenadas con la intención de lograr un producto de software de calidad. El proceso desarrollado considera que las necesidades del usuario deben ser traducidas en requerimientos de software, y estos luego transformados en diseño y el diseño implementado en código. Posteriormente, el código es probado, documentado y certificado para su uso operativo.

REVISIONES

Primera Revisión:

- Objetivos
- Revisión de la literatura

Segunda Revisión

- Metodología del trabajo
- Análisis de los requerimientos
- Diseño del programa
- Implementación

Tercera Revisión:

- Periodo de prueba
- Conclusiones y recomendaciones

NOTA

Extensión máxima: 100 páginas.



ÍNDICE DE CONTENIDO

	Página
CAPÍTULO 1: INTRODUCCIÓN	1
1.1. Objetivos	1
1.1.1. Objetivo general	1
1.1.2. Objetivos Específicos	1
1.2. Alcance del proyecto	2
1.3. Limitaciones	3
CAPÍTULO 2: REVISIÓN DE LITERATURA	4
2.1. Del tránsito a la movilidad sostenible	4
2.1.1. Tránsito	5
2.1.2. Transporte público	6
2.1.3. Movilidad	7
2.1.4. Movilidad sostenible y accesibilidad sostenible	8
2.2. Espacio público	9
2.3. Vida Pública	11
2.4. Estudios de vida pública	13
CAPÍTULO 3: METODOLOGÍA	19
3.1. Etapas de desarrollo de software	19
3.1.1. Análisis de requerimientos	20
3.1.2. Diseño	20
3.1.3. Implementación	21
3.1.4. Prueba y mantenimiento	22
3.1.5. Documentación	22
3.2. Modelos de desarrollo de software	23
3.2.1. Modelo de cascada	23
3.2.2. Modelo prototipado	24
3.2.3. Modelo incremental	25
CAPÍTULO 4: ANÁLISIS DE REQUERIMIENTOS	27
4.1. Requerimientos Funcionales	28
4.1.1. Pantalla Principal	28
4.1.2. Herramienta Tracing	29
4.1.3. Herramienta Mapping	30
4.1.4. Herramienta Tracking	31
4.1.5. Herramienta Counting	31
4.1.6. Herramienta Speed Test	32
4.1.7. Herramienta Diary	33
4.2. Casos de uso	34
4.2.1. Pantalla principal	34
4.2.2. Herramienta Tracing	36
4.2.3. Herramienta Mapping	40
4.2.4. Herramienta Tracking	41
4.2.5. Herramienta Counting	43
4.2.6. Herramienta Speed Test	45
4.2.7. Herramienta Diary	46

4.3. Requerimientos no funcionales	47
CAPÍTULO 5: DISEÑO DEL PROGRAMA	48
5.1. Interfaz gráfica de usuario (GUI)	48
5.1.1. Pantalla Principal	51
5.1.2. Herramienta Tracing	53
5.1.3. Herramienta Mapping	55
5.1.4. Herramienta Tracking	57
5.1.5. Herramienta Counting	59
5.1.6. Herramienta Speed Test	62
5.1.7. Herramienta Diary	65
5.2. Funciones utilizadas	67
5.2.1. Ingreso de datos	67
5.2.2. Manipulación de datos	69
5.2.3. Visualización de datos	70
5.3. Organización de los archivos	71
5.4. Estándares de código	71
CAPÍTULO 6: IMPLEMENTACIÓN	74
6.1. Construcción	74
6.1.1. Modelo de arquitectura	74
6.1.2. Lenguaje de programación	75
6.2. Pruebas Unitarias	77
6.2.1. Pantalla Principal	77
6.2.2. Herramienta Tracing	78
6.2.3. Herramienta Mapping	79
6.2.4. Herramienta Tracking	80
6.2.5. Herramienta Counting	81
6.2.6. Herramienta Speed Test	82
6.2.7. Herramienta Diary	83
CAPÍTULO 7: PERIODO DE PRUEBAS	85
7.1. Proyecto Estudio de vida pública y microsimulación en la calle Marqués de la ciudad de Cusco	85
7.2. Proyecto Estudio de vida pública y microsimulación peatonal en la vía PUCP	89
CAPÍTULO 8: CONCLUSIONES Y RECOMENDACIONES	92
8.1. Conclusiones	92
8.2. Recomendaciones	94
REFERENCIAS	95

Lista de Figuras

	Página
Figura 3.1. Esquema del modelo en cascada.	24
Figura 3.2. Esquema del modelo prototipado.	25
Figura 3.3. Esquema del modelo incremental.	25
Figura 5.1. Herramienta GUIDE de MATLAB	49
Figura 5.2. Elemento botón (Push Button)	49
Figura 5.3. Elemento barra deslizador (Slider)	49
Figura 5.4. Elemento menú desplegable (Pop-up Menu)	50
Figura 5.5. Elemento botón radial (Radio Button)	50
Figura 5.6. Elemento texto estático (Static Text)	50
Figura 5.7. Elemento texto editable (Edit Text)	50
Figura 5.8. Elemento panel(Panel)	50
Figura 5.9. Elemento Ejes (Axes)	51
Figura 5.10. Interfaz Pantalla Principal	52
Figura 5.11. Interfaz Tracing	53
Figura 5.12. PDF generado por la herramienta Tracing.	54
Figura 5.13. Interfaz de la herramienta Mapping.	55
Figura 5.14. PDF generado por la herramienta Mapping.	57
Figura 5.15. Interfaz de la herramienta Tracking.	58
Figura 5.16. PDF generados por la herramienta Tracking.	59
Figura 5.17. Interfaz de la herramienta Counting.	60
Figura 5.18. Ventana para ingresar datos – Herramienta Counting	61
Figura 5.19. PDF generados por la herramienta Counting	62
Figura 5.20. Interfaz de la herramienta Speed Test.	63
Figura 5.21. Ventana para ingresar datos – Herramienta Speed Test.	63
Figura 5.22. PDF generados por la herramienta Counting.	64
Figura 5.23. Interfaz de la herramienta Diary.	65
Figura 5.24. PDF generados por la herramienta Diary.	66
Figura 5.25. Ventana herramienta uipickfiles.	68
Figura 6.1. Interfaz de la ventana Editor de MATLAB	76
Figura 7.1. Interfaz herramienta Mapping	86
Figura 7.2. Interfaz herramienta Tracing	86
Figura 7.3. Interfaz herramienta Tracking	87
Figura 7.4. Interfaz herramienta Counting	87
Figura 7.5. Interfaz herramienta Speed Test	88
Figura 7.6. Interfaz herramienta Diary	88
Figura 7.7. Interfaz herramienta Mapping	90
Figura 7.8. Interfaz herramienta Counting	90
Figura 7.9. Interfaz herramienta Speed Test	91
Figura 7.10. Interfaz herramienta Diary	91

Lista de Tablas

	Página
Tabla 4.1. Caso de uso Activar un proyecto.	35
Tabla 4.2. Caso de uso Salvar un proyecto.	35
Tabla 4.3. Caso de uso Modificar información del proyecto.	36
Tabla 4.4. Caso de uso Modificar información del registro.	37
Tabla 4.5. Caso de uso Crear un trazo.	38
Tabla 4.6. Caso de uso Administrar trazos.	39
Tabla 4.7. Caso de uso Exportar datos.	39
Tabla 4.8. Caso de uso Administrar actividades.	40
Tabla 4.9. Caso de uso Editar actividad.	41
Tabla 4.10. Caso de uso Definir trayectoria.	42
Tabla 4.11. Caso de uso Ingresar registros.	43
Tabla 4.12. Caso de uso Definir datos del registro.	44
Tabla 4.13. Caso de uso Ingresar datos.	44
Tabla 4.14. Caso de uso Definir secciones de análisis.	45
Tabla 4.15. Caso de uso Ingresar datos.	45
Tabla 4.16. Caso de uso Editar registros.	46
Tabla 6.1. Resumen de resultados de pruebas unitarias – Pantalla Principal	77
Tabla 6.2. Resumen de resultados de pruebas unitarias – Herramienta Tracing	78
Tabla 6.3. Resumen de resultados de pruebas unitarias – Herramienta Mapping	80
Tabla 6.4. Resumen de resultados de pruebas unitarias – Herramienta Tracking	81
Tabla 6.5. Resumen de resultados de pruebas unitarias – Herramienta Counting	82
Tabla 6.6. Resumen de resultados de pruebas unitarias – Herramienta Speed Test.	83
Tabla 6.7. Resumen de resultados de pruebas unitarias – Herramienta Diary.	84

CAPÍTULO 1: INTRODUCCIÓN

En los últimos años, arquitectos y urbanistas han caído en la cuenta de que la forma en la que las ciudades crecían había provocado el deterioro de la calidad de vida de sus habitantes. El paradigma que planteaba la segregación entre vehículos motorizados y los demás usuarios como peatones y ciclistas, dando prioridad a los primeros, ha dejado de ser considerado válido en varias partes del mundo, sobre todo en Europa. Ciudades como Barcelona y Copenhague han enfocado sus esfuerzos en el desarrollo de lo que hoy en día se conoce como espacio y vida pública. Es importante definir estos dos conceptos diferentes pero que se encuentran en constante interacción. El primero se refiere al entorno físico y engloba las calles, parques, edificios, etc. El segundo concepto es mucho más difícil de delimitar, la vida pública todo aquello que acontece alrededor y dentro del espacio público (Gehl & Svarre, 2013).

¿Cómo es posible diseñar parte del espacio público si no se conoce cómo funciona la vida pública? Es bajo esta premisa que, según Gehl (2013), se fundamentan los Estudios de Vida Pública. Éstos buscan analizar cómo las personas se desenvuelven en diferentes entornos para que, en base a los resultados, pueda diseñarse el espacio público. En su libro, Gehl & Svarre enlistan herramientas sencillas que permiten recolectar datos de campo y que hoy en día son utilizadas. Sin embargo, actualmente no existe una herramienta que permita almacenar y procesar los datos recogidos en un formato estándar. El presente trabajo busca atender dicha carencia al desarrollar un programa que permita el ingreso y la visualización de los datos recolectados en los estudios de vida pública.

1.1. Objetivos

1.1.1. Objetivo general

El objetivo general del proyecto es desarrollar una herramienta informática, basada en MATLAB, y con una interfaz gráfica de usuario, que permita procesar, de manera estándar, los datos recolectados mediante las herramientas básicas utilizadas en los estudios de vida pública.

1.1.2. Objetivos Específicos

Desarrollar una herramienta informática en MATLAB que cuente con una interfaz gráfica sencilla y que no requieran conocimientos de programación para ser utilizada. Este programa permite el ingreso,

almacenamiento y visualización de la información recolectada a través de las herramientas básicas empleadas en los estudios de vida pública. Las herramientas implementadas son mapeo, trazado, seguimiento, conteo, test de velocidad y registro fotográfico/escrito. Con este programa se busca proponer una alternativa que estandarice la toma de datos para facilitar su análisis y entendimiento.

1.2. Alcance del proyecto

El proyecto propone diseñar e implementar una herramienta informática sencilla que proponga una manera estándar de almacenar y visualizar los datos recolectados en los estudios de vida pública. Para el desarrollo del programa se usó el software y lenguaje de programación MATLAB. El programa cuenta con su propia interfaz gráfica con el fin de que su uso sea intuitivo para todos los usuarios, aunque no cuenten con conocimientos sobre programación o MATLAB.

Los estudios de vida pública cuentan con distintas herramientas de las cuales el proyecto implementa seis. La herramienta de mapeo permite identificar y ubicar actividades de interés sobre un croquis del espacio analizado. Con el trazado se identifica las rutas, también llamadas líneas de deseo, descritas por las personas dentro del espacio público. Cuando el espacio analizado es muy extenso, la herramienta de seguimiento registra detalladamente la ruta de una sola persona en particular a lo largo de todo su recorrido. El conteo permite ingresar datos cuantitativos de la cantidad de personas que atraviesan una sección delimitada por el usuario en intervalos de tiempo constantes. Los test de velocidad registran las velocidades empleadas por distintos peatones para movilizarse entre dos secciones definidas por el usuario. Finalmente, el registro fotográfico y escrito a través de un diario permite identificar detalles que escapan al alcance de las demás herramientas. Todos estos datos también son clasificados de acuerdo a criterios de sexo, edad y la presencia de discapacidades en los usuarios estudiados.

Para administrar la información, el programa trabaja en base a proyectos. El usuario puede crear un proyecto en el cual se guardan los datos que ingrese a través de las distintas herramientas. Posteriormente, estos proyectos pueden ser exportados y compartidos con otros usuarios. Además de la información asociada a cada herramienta, también es posible registrar información global como la fecha en la que se realizó el estudio, el lugar analizado, el nombre del

usuario y las condiciones climatológicas. Finalmente, el programa permite generar reportes en formato PDF que muestran los datos ingresados y pueden ser compartidos sin necesidad de que el receptor cuente con el programa.

1.3. Limitaciones

El programa fue desarrollado en el entorno y lenguaje de programación MATLAB, por lo que para ser utilizado los usuarios deberán contar con un compilador o con la versión completa de MATLAB. Por ello, la guía de usuario incluirá el procedimiento para ejecutar el programa desde un compilador gratuito.

Un punto importante a resaltar es que MATLAB no es el lenguaje ideal para este tipo de proyecto. Existen alternativas como Java o Python que permiten elaborar programas mucho más visuales e interactivos. MATLAB es un lenguaje de programación potente cuando se desea implementar operaciones matemáticas complejas y/o pesadas como análisis estadísticos, tratamiento de señales, entre otros. A pesar de lo mencionado se decidió utilizar el lenguaje MATLAB en base a la experiencia y conocimientos del estudiante que implementó el proyecto.

Finalmente, el programa no cuenta con todas las funcionalidades con las que la mayoría de softwares comerciales posee hoy en día ya que el estudiante no cuenta con experiencia ni capacitación previa en desarrollo de software. Por ejemplo, no será posible retroceder en las acciones realizadas (comando Ctrl + Z). Por ello el usuario deberá utilizar el programa siguiendo las indicaciones del desarrollador para evitar el mal funcionamiento y errores no previstos. Las instrucciones necesarias para utilizar las funcionalidades disponibles son detalladas en la guía de usuario del programa.

CAPÍTULO 2: REVISIÓN DE LITERATURA

Como se mencionó en el capítulo anterior, el objetivo del presente proyecto es desarrollar una herramienta para el almacenamiento de los datos recolectados en estudios de vida pública o también llamados estudios de vida urbana. Para poder comprender la importancia y utilidad de esta herramienta es necesario repasar algunos conceptos más generales. Las herramientas que el programa implementa fueron formuladas con base en la compleja naturaleza que caracteriza a la vida pública. Sin embargo, no se puede hablar de vida pública sin mencionar el espacio público ya que es en éste donde se desarrolla. En este punto, para poder definir los conceptos de estudio y vida pública no se puede pasar por alto el planeamiento urbano y su evolución en la última década: partiendo del tránsito de vehículos particulares, luego pasa por el transporte público masivo para llegar al concepto de movilidad y finalmente se complementa con el concepto de sostenibilidad (Dextre y Avellaneda, 2014).

El objetivo del presente capítulo es presentar un marco teórico a partir de la literatura revisada. El orden de presentación de los conceptos ira en sentido opuesto a la enumeración enlistada líneas arriba. Se parte desde los conceptos más amplios asociados al planeamiento urbano para llegar a las herramientas de los estudios de vida pública.

2.1. Del tránsito a la movilidad sostenible

Desde la década de 1960, las políticas destinadas a aliviar la congestión vehicular se han enfocado en aumentar la infraestructura destinada a los vehículos motorizados. No obstante, se ha demostrado que estas aparentes soluciones no son sostenibles en el tiempo, ya que, si bien alivian el problema a corto plazo, no llegan a resolverlo. Estudiosos como Jane Jacobs y William Whyte en Nueva York, Christopher Alexander en Berkeley y Jan Gehl en Copenhague criticaron la forma en la que las ciudades crecían. Con el paso de los años, diversos autores han ido desarrollando diversos enfoques que estudian la problemática del desplazamiento. En un primer paso, el concepto de tránsito, que se limitaba al automóvil particular, evolucionó para incluir el transporte público masivo dada su mayor eficiencia en términos de personas movilizadas por hora.

Sin embargo, las soluciones enfocadas en el transporte motorizado tampoco llegaron a solucionar el problema ya que perdían de vista a los usuarios no motorizados. A partir de las necesidades de estos últimos nace el concepto de

movilidad. Finalmente, cuando la atención no solo se fija en los desplazamientos, sino también se toma en consideración el impacto que estos generan en el ambiente, se llega al concepto de movilidad sostenible que, actualmente, es uno de los enfoques más completos usados para el planeamiento urbano.

2.1.1. Tránsito

Durante muchos años, el tránsito ha sido la disciplina encargada del diseño de ciudades. Según Radelat (2003), el objetivo del tránsito es la circulación segura y eficiente de peatones y vehículos. En el caso de los últimos sin importar la cantidad de personas que traslade o los objetos que transporte mientras no representen peligro. Por ello, el tránsito mide la eficiencia a través de la cantidad de vehículos que circulan por una determinada vía por unidad de tiempo.

De acuerdo con esta definición, las ciudades se han esforzado en construir infraestructura priorizando los vehículos motorizados. No obstante, la evidencia demuestra que estructuras como pasos a desnivel y autopistas producen un efecto de segregación en la zona donde son construidas. Mientras que se aumenta la velocidad de circulación para los automóviles, para los residentes y personas que realizan sus actividades en la zona se produce un efecto barrera que limita sus actividades. Este efecto es más notable cuando se trata de usuarios vulnerables como niños, ancianos y discapacitados.

Sin ir muy lejos, se puede analizar el caso peruano con respecto a la infraestructura vial. Concretamente en la capital, la evidencia demuestra que la construcción de obras como la vía expresa en la avenida Javier Prado, el paso a desnivel de la avenida Perú sobre la avenida Universitaria o el nuevo bypass en la avenida 28 de Julio se han convertido en inversiones millonarias que no han solucionado el problema del tráfico. Por otro lado, si se analiza los puentes peatonales, con base en la información recolectada por Dextre y Avellaneda (2014), la cantidad de accidentes de tránsito es mayor en comparación a ciudades que carecen de ellos.

La analogía presentada por Jiménez (2013) ilustra la problemática del tránsito. Si el parque automotor es un gas que está contenido en un recipiente cerrado, que vendría a ser la actual infraestructura vial, al

aumentar la cantidad de gas aumenta la presión en el recipiente por lo que tarde o temprano será necesario cambiarlo por uno más grande, es decir construir más obras como pasos a desnivel y autopistas. Sin embargo, mientras el gas siga aumentando y vuelva a llenar el nuevo recipiente, será necesario cambiarlo por uno aún más grande. De esta sencilla comparación se concluye que la solución al problema no consiste en aumentar la capacidad cada vez que se satura la existente; es necesario ir más allá y solucionar el problema de trasfondo, evitar que se siga generando gas o, volviendo al contexto, la cantidad de vehículos motorizados.

2.1.2. Transporte público

Con el paso de los años y con base en la experiencia de las grandes ciudades, principalmente en Estados Unidos, se actualizó la forma de medir la eficiencia del transporte. Lo que antes era medido según la cantidad de vehículos movilizadas se tradujo en mayor cantidad de personas movilizadas, mejor uso del espacio y menor contaminación ambiental. Estos parámetros se ven optimizados cuando se emplean los medios de transporte público masivos.

Por ello, basándose en la cantidad de personas que se necesita movilizar, se opta por distintos medios de transporte público masivo. Según la información presentada por Dextre y Avellaneda (2014), el Trolebús de Quito tiene una capacidad de nueve mil personas por hora. Un tranvía puede llegar a transportar hasta once mil quinientas personas por hora. Para demandas más grandes, de hasta treinta mil personas por hora, se emplea un sistema de buses de tránsito rápido (BRT por sus siglas en inglés: Bus Rapid Transit) con carriles exclusivos, como El Metropolitano de la ciudad de Lima. Finalmente, cuando la demanda supera las cuarenta mil personas por hora es necesario un sistema de metro.

Los sistemas de transporte público masivo reducen los tiempos de viaje y aumentan la cantidad de personas movilizadas; sin embargo, las distancias de viaje permanecen iguales por lo que no se soluciona el problema por completo. A largo plazo, como los tiempos de viaje son menores, se construirán centros de actividades más lejanos y se volverá al estado inicial. Por ello, si bien el transporte público masivo es

fundamental para satisfacer las necesidades de desplazamiento, esta solución sigue siendo de carácter técnico y es tomada según la cantidad de personas que se desea movilizar.

2.1.3. Movilidad

La movilidad es una versión más inclusiva del concepto de tránsito y que se utiliza en el contexto de la planificación urbana. La diferencia sustancial entre tránsito y movilidad es que, mientras el primero se enfoca en la circulación de vehículos motorizados, la movilidad amplía el objeto de estudio y añade el movimiento de personas y mercancías sin tener en cuenta la jerarquía del motor (Sanz, 2010). Aunque para muchas personas e instituciones el cambio ha sido únicamente de carácter nominal.

El nuevo sujeto de estudio incluye a las personas que se movilizan a pie, en bicicleta y, en general, cualquier vehículo no motorizado. Las características que antes eran encasilladas dentro de la categoría “conductor estándar” ahora son analizadas de manera independiente. Por ejemplo, la edad, sexo, clase social, grupo étnico, condición física o psíquica y cualquiera otra que permita identificar las necesidades de los usuarios.

Según Tonucci (2004) la ciudad debe ser diseñada para los niños. De esta manera, si ellos pueden utilizar la ciudad de manera segura y cómoda entonces cualquier otro usuario debe poder hacerlo. Análogamente, Tayler (2002) propone basarse en las necesidades de los discapacitados para garantizar que todos los usuarios pueden desplazarse de forma cómoda y segura.

Como resulta obvio, si tanto el usuario y el objeto de estudio han sido modificados, las herramientas empleadas para estudiarlos dejan de ser aplicables y también deben ser actualizadas. Ahora existen múltiples dimensiones en lo que antes podía ser modelado como un fenómeno físico, como la asimilación del movimiento de automóviles al flujo de un fluido o el modelo de seguimiento vehicular. Los conocimientos propios de la ingeniería civil o la arquitectura no son suficientes y es necesario incluir la perspectiva de la psicología, la sociología, entre otras especialidades. Esto convierte a la planificación urbana en una tarea multidisciplinaria que, bajo un nuevo conjunto de lentes, debe analizar las

necesidades de todos los usuarios que forman parte del universo de estudio.

2.1.4. Movilidad sostenible y accesibilidad sostenible

Las consideraciones con respecto al impacto del medio ambiente que nacen con base en el tránsito también han asumido una definición más extensa: la sostenibilidad. Este nuevo concepto atiende el impacto ambiental y socioeconómico de las actividades humanas (Sanz, 2010).

El último escalón en la evolución de la planificación urbana se alcanza cuando se combina el concepto de movilidad con el concepto de sostenibilidad. Desde este nuevo nivel, no basta preocuparse únicamente por el movimiento de las personas y mercancías, sino también en el impacto producido en el medio. Bajo esta perspectiva, el ordenamiento territorial cobra mayor importancia ya que las necesidades de desplazamiento dependen de la estructura de la ciudad.

Para poder reducir el impacto ambiental de los desplazamientos, el transporte público resulta la mejor opción en comparación a los vehículos particulares gracias a su eficiencia en términos de cantidad de personas por hora. Sin embargo, el sistema de transporte público debe ser multimodal con el fin de garantizar que todos los usuarios tengan acceso a él. Esto significa que deben existir distintos medios de transporte en la ciudad que deben trabajar en conjunto y no competir unos contra otros.

Dextre (2014) presenta un perfecto ejemplo que ilustra la importancia de un sistema multimodal. Para un usuario de avanzada edad, utilizar el sistema de metro puede resultar difícil si no cuenta con otro medio de transporte que lo pueda movilizar hacia las estaciones del tren, que puede tener sus estaciones cada 500 metros o más. En este caso es necesario integrar otro medio de transporte como por ejemplo los buses. Este ejemplo demuestra que los conceptos de accesibilidad y movilidad sostenible son complementarios y que funcionan a distintas escalas.

Cuando una ciudad descentraliza sus centros de actividades, como centros educativos, laborales, de salud o recreación, se denomina policéntrica. Este es el modelo de ciudad que busca la movilidad sostenible ya que, optimiza el ordenamiento territorial. Si las personas cuentan con lugares más cercanos a sus viviendas en donde pueden

satisfacer sus necesidades, se reduce la necesidad de viajar largas distancias, Sanz (2009). Es importante señalar que se habla de reducir la necesidad, mas no eliminarla, ya que es posible que las personas aún prefieran viajar largas distancias hacia otros centros de la ciudad. Lo que si se garantiza es que las personas ya no están condicionadas a moverse hacia un único punto para satisfacer sus necesidades. Cabe señalar que, para que este modelo funcione los servicios brindados en todos los centros deben ser accesibles y de calidad.

En el escenario de una ciudad policéntrica, las personas pueden moverse dentro de su entorno cercano o entre distintos centros de la ciudad. A nivel micro, las personas viajan distancias más cortas por lo que pueden moverse a pie o en bicicleta. En este nivel se debe garantizar la accesibilidad a todos los servicios para todos los usuarios. Por otro lado, en una escala macro, cuando se trata de desplazarse largas distancias, la movilidad sostenible debe garantizar un sistema de transporte público eficiente.

Como se ha presentado, el planeamiento urbano ha ido evolucionando a medida que se iban desarrollando distintos enfoques. No obstante, el objetivo siempre ha sido el mismo: garantizar a los ciudadanos las condiciones favorables para el desarrollo de su vida y trabajo (Buchanan, 1964). Teniendo en cuenta dicho propósito, el planeamiento urbano busca modificar el espacio público para satisfacer las necesidades de las personas. Éstas no se limitan únicamente al desplazamiento ya que también existen necesidades de carácter social propias del concepto de espacio y vida pública.

2.2. Espacio público

El concepto de espacio público varía según diversos autores y el contexto en el que se estudian debido a que tiene una naturaleza polisémica (Ludeña, 2013). Sin embargo, es inevitable desligarlo de su dimensión espacial ya que el espacio público debe ser, como su nombre indica, un espacio físico. Dentro del contexto del libro *How to Study Public Life* de Jan Gehl (2013), el autor define, de manera muy genérica, el espacio público como el conjunto de calles, edificios, parques y todo aquello que es considerado parte del entorno construido.

Según esta definición, nacen las interrogantes ¿Cuáles son los límites físicos del espacio público? ¿Este espacio se limita únicamente a las áreas libres,

excluyendo los ambientes cerrados o techados? ¿Qué condiciones son necesarias y cuáles son suficientes para denominar a un espacio como público?

Según el Reglamento Nacional de Edificaciones (Ministerio de Vivienda, 2006) el espacio público es la superficie de uso público, destinada a circulación o recreación. Más adelante, en la modificación del reglamento del año 2011, en el artículo 1 de la Norma GH 020, se incluyen las áreas destinadas a parques y plazas de uso público, así como las vías de circulación vehicular y peatonal a la definición antes citada. Esta definición parece limitar el concepto sólo al “espacio libre de uso público” y excluye todo espacio que forma parte de la propiedad privada la cual, según otros autores, también forma parte del espacio público.

Si se revisa la normativa colombiana relacionada al espacio público (Presidencia de la República de Colombia, 1998) se encuentra una definición más extensiva en la cual se define que:

El espacio público comprende, entre otros, los siguientes aspectos: [1] Los bienes de uso público, es decir aquellos inmuebles de dominio público cuyo uso pertenece a todos los habitantes del territorio nacional, destinados al uso o disfrute colectivo; [2] los elementos arquitectónicos, espaciales y naturales de los inmuebles de propiedad privada que por su naturaleza, uso o afectación satisfacen necesidades de uso público; y [3] las áreas requeridas para la conformación del sistema de espacio público en los términos establecidos en este Decreto.

A diferencia de la normativa peruana, se incluye al dominio del espacio público los espacios cerrados y/o de propiedad privada como por ejemplo centros comerciales y bibliotecas.

Aunque las definiciones parezcan contradictorias, para Ludeña (2013) ambas son correctas con la diferencia que la perspectiva colombiana es más incluyente y extensiva. Lo que se desprende de esta aparente contradicción es lo complejo que resulta delimitar el concepto de espacio público cuando solo se considera su dimensión física, más aún cuando esta no es la más importante.

Para llegar a una mejor definición, Vega Centeno (2006) sugiere partir de los usos que la comunidad le da al espacio público diariamente. Para él, los espacios públicos son los espacios que permiten asegurar un futuro sostenible

para la comunidad en la medida que aseguran la pluralidad de encuentros y actividades de los usuarios. De esta definición se puede desprender una característica importante: el uso del espacio público es compartido por lo que se debe respetar las actividades de todos los usuarios.

Siguiendo la misma línea, Borja (2003) sostiene que el espacio público supone un dominio público de uso social. Desde esta perspectiva, los usos pueden ser muy variados con base en las necesidades particulares de los usuarios. Es por ello que se suele hablar del carácter multifuncional del espacio público. No obstante, esta definición parece limitar los usos a necesidades que se entienden como actividades obligatorias; lo que excluye las actividades opcionales y, con ello, una de las características más importantes del espacio público: la libertad. Por ello, la definición que plantean Remy y Voyé (1981) resulta mucho más adecuada: el espacio público es todo espacio accesible, sin importar el momento o tipo de personas, y sin discriminación alguna sobre las actividades realizadas.

Para finalizar, la definición de Ludeña (2013) añade una última característica que resulta interesante: el espacio público no es “espacio público” porque sea utilizado por el “público”, sino porque en su uso; ya sea físico, mental o emocional; reproduce un determinado valor de lo público. Este concepto tiene como punto de partida y de fin al ser humano, a nivel individual y colectivo, como elemento indispensable. Para que un lugar sea catalogado como espacio público no es requisito que las actividades humanas lo ocupen de manera física, sino también puede ser subjetiva. Por ello, existen espacios públicos que tienen un valor paisajístico, que están dedicados a la contemplación y que nunca llegan a ser ocupados por las personas.

De entre todas las definiciones revisadas, existe un punto en común: el desarrollo de actividades humanas dentro del espacio público. Por ello, cuando se busca rediseñar la ciudad, y por consiguiente el espacio público, es necesario tomar en cuenta las necesidades, no solo de desplazamiento, sino también de carácter social de la comunidad. A partir de este punto, cuando la atención se fija en las actividades y no el entorno físico en el que se desarrollan, se empieza a hablar de vida pública.

2.3. Vida Pública

Gehl y Svarre (2013) definen la vida pública, en su sentido más amplio, como todas las actividades que las personas desarrollan dentro del espacio público.

De manera más concreta, se puede decir que la vida pública se compone del comportamiento humano de carácter individual y colectivo. A este nivel, es importante señalar que la forma en la cual las personas actúan cuando se encuentran en un espacio público, es muy diferente a su forma de actuar en un ambiente privado. Otra acotación importante es que la vida pública no define el llamado “buen comportamiento” que las personas deberían seguir.

Si bien la vida humana debe resultar natural, es imposible que se desarrolle de manera adecuada si el entorno, el espacio público, no lo permite. Esta característica resalta en la estructura de las ciudades antiguas. En éstas, el crecimiento era pausado y avanzaba acorde a las necesidades de la población como el comercio. Sin embargo, con el avance de la tecnología, el crecimiento de las ciudades se vio acelerado y se perdieron muchas costumbres. Según Jacobs (1961) el crecimiento de las ciudades enfocado en el automóvil degeneró el espacio público y acorraló la vida pública hasta el punto de extinguirla. Debido a este fenómeno, como se mencionó en este capítulo, nacieron corrientes que buscaban renovar la forma de diseñar las ciudades y, con ello, revalorar la importancia de la vida pública. La propuesta no busca regresar a las prácticas de antaño, sino desarrollar herramientas que permitan armonizar el espacio y vida pública (Gehl y Svarre, 2013),

Con este objetivo, es crucial analizar y predecir el comportamiento de las personas; no obstante, esta tarea resulta complicada. Esta complicación se debe a que la característica más importante de la vida pública es su complejidad ya que depende de diversos factores. A nivel del entorno influyen el lugar, tiempo y condiciones climáticas. Las características de las personas como edad, sexo y la presencia de discapacidades también condicionan el comportamiento. Finalmente, deben considerarse los intereses y objetivos particulares de cada uno.

Gehl (1968), en un estudio para correlacionar el espacio público y la vida pública, propone clasificar las actividades según el grado de necesidad que representen. En los extremos de esta escala se encuentran las actividades necesarias y las opcionales. En primer lugar, se define las “actividades necesarias” como aquellas que deben ser realizadas sin importar las condiciones externas y que atienden a un requerimiento de la persona. Entre estas se encuentra caminar hacia el trabajo o esperar el transporte público. El segundo grupo está compuesto por las “actividades opcionales” que se

desarrollan cuando las condiciones externas, como el clima o la hora, son ideales y el usuario lo desea. Actividades como salir a pasear y sentarse en una plaza a tomar el sol no demandan una necesidad urgente y su no realización no perjudica a la persona directamente. Ambas categorías no son rígidas ya que varían según la persona y las circunstancias ya que, actividades que para algunos son necesarias pueden ser opcionales para otros.

Desde otro criterio, se puede definir las “actividades sociales” como aquellas que involucran la interacción, no necesariamente directa, entre dos o más personas. Estas actividades no se desarrollan únicamente con personas conocidas. Uno puede conocer a personas que vivan cerca a su casa o que frecuenten los mismos lugares. En este contexto, William Whyte (1980) llama “triangulación” el fenómeno donde un estímulo externo da pie a que dos personas que no se conocen empiecen a conversar. De esta forma, se ratifica el rol del espacio público como lugar de encuentro.

Para diseñar un espacio público que permita satisfacer las necesidades de la comunidad en todas sus dimensiones. Dicho de otra manera, "el núcleo del asunto es obtener la mayor cantidad de vida en el espacio público que funcionen de tal manera que permita a la vida cotidiana desarrollarse bajo condiciones decentes y cooperar con el entorno físico en lugar de luchar contra él" (Gehl y Sverre, 2013).

2.4. Estudios de vida pública

Al momento de fijar la atención en la vida pública las preguntas que se pueden formular al respecto son prácticamente interminables. Por ejemplo, cuando se observa un grupo de personas sentadas en el césped de un parque, además de preguntar dónde están sentadas también urge preguntar ¿quiénes son? ¿cuántos años tienen? ¿cuánto tiempo llevan sentados? ¿por qué eligieron ese lugar en particular? ¿están sentados porque esperan algo, para descansar o existe otro motivo? y así la lista de preguntas puede continuar. Por más específicas que pudiesen parecer, todas brindan información útil para entender el comportamiento de las personas.

Las respuestas que se obtengan, así como las preguntas que se necesiten formular, podrán ser completamente diferentes dependiendo de la estación del año, el sexo de las personas, la hora o el lugar analizado. Con esto nuevamente queda expuesta la complejidad inherente a la vida pública. Por

ello, es necesario prestar atención no solo al hecho en sí, sino también al objetivo de la actividad y la cadena de eventos en que se desarrolla.

Dado que las respuestas pueden ser tan variadas se prefiere empezar con preguntas básicas que brinden un vistazo general de la situación analizada. Gehl y Svarre (2013) proponen cinco.

¿Cuántos? Es posible registrar la cantidad de personas que desarrollan una actividad en particular. Esta pregunta provee una respuesta cuantitativa que, por su naturaleza, resulta más fácil de interpretar y comparar. Para ello, sin importar lo que se cuente, es necesario realizar varios registros ya que un valor individual no brinda información representativa. Cuando se tiene múltiples valores es posible realizar análisis estadísticos sencillos para estimar el valor esperado o la dispersión de la variable contabilizada. Por ejemplo, se puede comparar la cantidad de personas que utilizan una ruta antes y después de realizar una modificación y, de esta manera, evaluar la efectividad de la intervención realizada.

¿Quién? Cuando se habla de vida pública se refiere al comportamiento de las personas en general. Sin embargo, el término personas engloba distintos grupos de características variables. Estos grupos pueden ser clasificados bajo criterios simples, como el sexo y la edad, y otro más complejos como la presencia de discapacidades o restricciones. De esta manera se puede analizar cada grupo de manera independiente e identificar sus necesidades particulares. Esta información es muy importante cuando se desea concentrar los esfuerzos en los usuarios vulnerables, como ancianos y discapacitados. Criterios más complejos como la situación socioeconómica o el rubro laboral resultan más difíciles de percibir en base a la observación.

¿Dónde? Una vez delimitado el espacio analizado, es importante ubicar, y predecir la ubicación, en la cual se desarrollan las distintas actividades. Este punto es importante ya que, al momento de diseñar el espacio público, se espera que las personas le den el uso esperado. Sin embargo, dicho comportamiento no siempre es el esperado por lo que los espacios destinados para ciertas actividades pueden no cumplir con su objetivo. Respondiendo a esta pregunta se puede ubicar los lugares donde las personas prefieren descansar o la dirección de los flujos de peatones.

¿Qué? Definir qué ocurre dentro de un espacio permite formar un panorama y categorizar las actividades que ocurren, por ejemplo, actividades comerciales o de entretenimiento. La cantidad de actividades que se puede observar es prácticamente ilimitada. Por ello, es importante definir actividades básicas como caminar, detenerse o sentarse; o definir categorías que engloben distintas actividades y faciliten su clasificación. Un criterio útil es el mencionado en este capítulo con base en el grado de necesidad de cada actividad. La información asociada a esta pregunta resulta de vital importancia ya que permite comprender el uso que le dan los usuarios al espacio público y la función que este cumple. Las actividades pueden variar dependiendo de la persona observada, las condiciones climáticas, la hora del día, entre otros factores.

¿Cuánto tiempo? La última dimensión analizada, mas no la menos importante, es el tiempo. A diferencia de estudiar elementos estáticos como los edificios, las personas se encuentran en constante cambio. El comportamiento de una persona puede ser completamente diferente al registrado minutos atrás. Por un lado, se puede registrar cuánto tiempo toma a las personas completar actividades necesarias como desplazarse de un punto a otro. Por otro lado, se puede medir el tiempo que una persona decide realizar una actividad opcional. En base a estos resultados se puede medir la calidad del espacio público, identificar los elementos que producen demoras o aquellos que hacen más comfortable la estancia de los usuarios.

Luego de definir qué es lo que se desea estudiar, queda proponer la forma en la que estos datos serán recolectados. Diversos autores coinciden que la principal herramienta más eficiente para identificar y registrar la vida pública es la observación. De manera similar al estudio de especies animales en su hábitat, el observador debe ser lo más neutral posible y actuar como un ente invisible para mantener una perspectiva global. Gehl y Svarre señalan que el “observar” no se debe limitar al sentido de la vista, el observador debe concentrar todos sus sentidos para captar hasta el más mínimo detalle y cómo “lo ordinaria se desenvuelve en el espacio público”.

Es importante señalar que la toma de datos no puede delegarse a un equipo electrónico. Si bien es posible emplear equipos para contabilizar flujos peatonales o medir velocidades, estos aparatos carecen de criterio para discernir la información. Dada la naturaleza compleja de la vida pública, es

necesario interpretar, en tiempo real, los datos registrados. De esta manera no solo se puede identificar patrones o comportamientos extraños sino también sus causas e impacto en el entorno. Además, para tener un mejor panorama es necesario realizar el estudio en distintas épocas del año, bajo distintas condiciones climáticas y en distintas horas del día.

Los únicos implementos que el observador necesita son lápiz y papel, con ellos se puede registrar información de muchas formas; prácticamente todo se puede contar, es posible dibujar los patrones que se identifican y finalmente redactar aquello que se observa. Existen múltiples formas de registrar la información para un estudio de vida pública. Estas metodologías serán denominadas herramientas. La elección de las herramientas depende de diversos factores como el espacio analizado, el presupuesto disponible, las condiciones locales, el objetivo del estudio, entre otras. De acuerdo a la magnitud del espacio analizado, ya que no es lo mismo si se estudia una plaza, o una calle, o un sector de la ciudad, se pueden preferir ciertas herramientas más que otras. Según la hipótesis que se desee probar se preferirán las herramientas que brinden información sobre las actividades de interés. Para definir cuándo se realizará la recolección de datos se debe tener en cuenta las condiciones climáticas y el contexto social y cultural del espacio: por ejemplo, se prefiere realizar el estudio bajo condiciones climáticas óptimas de acuerdo a la estación del año

Gehl y Svarre (2013) identifican ocho herramientas que pueden ser empleadas para estudiar la vida pública.

El mapeo permite ubicar a las personas según la actividad que estén realizando. Empleando puntos y marcas el observador puede señalar los lugares preferidos para tomar asiento, las áreas de mayor circulación y en general el uso de las zonas del espacio analizado.

El trazado sirve para identificar los movimientos de las personas en el espacio analizado. Mediante líneas se puede representar las líneas de deseo para identificar patrones o los lugares evitados.

Cuando el espacio analizado es muy extenso, el seguimiento se enfoca en una persona y registrar todas sus actividades dentro del espacio de interés. La persona puede ignorar el hecho de que está siendo seguido o se le puede informar al respecto antes.

La búsqueda de rastros permite identificar actividades que ocurrieron antes de la llegada del observador. Gracias al hecho de que la vida pública suele dejar marcas en el entorno no resulta muy difícil interpretarlas.

Mediante la fotografía es posible registrar información que de otra manera resultaría complicado y tedioso. En línea con el famoso refrán “una imagen vale más que mil palabras”, una fotografía muestra más de una actividad a la vez para poder analizarlas a detalle.

El registro de texto en un diario permite añadir información adicional que no encaja con las demás herramientas pero que a pesar de ello es importante. La ventaja de un diario es que prácticamente cualquier tipo de información puede ser añadida, incluyendo sus detalles.

Finalmente, los recorridos de prueba a través de zonas de interés permiten identificar el tiempo que toma a los usuarios desplazarse de un punto a otro. Además, permite reconocer elementos que obstaculizan o dificultan el tránsito de las personas.

Todas las herramientas deben ser combinadas con el criterio del observador. De esta forma, se puede clasificar los datos recolectados bajo criterios de sexo o rangos de edad. Una categoría que ha tomado importancia es la de los usuarios vulnerables, es decir niños, ancianos y discapacitados. Identificar sus necesidades resulta vital cuando se busca diseñar el espacio público bajo un enfoque inclusivo.

Llegado a este punto, es necesario volver a recalcar la importancia de los estudios de vida pública. ¿Cómo diseñar el espacio público si no se tiene claro los usos que la comunidad le da? De lo observado, se obtiene información variada: un desbalance entre la cantidad de hombres y mujeres puede representar un ambiente hostil o inseguro para un género, reducidas velocidad para cruzar un espacio pueden significar la presencia de elementos que producen demoras. La concentración de personas en solo ciertas zonas refleja la necesidad de replantear las áreas desocupadas. Según las distancias que recorren los peatones resulta necesario reubicar los paraderos del transporte público. Así, la información que se obtiene depende de la interpretación que se les dé a los datos.

De entre los datos recolectados, a diferencia de los conteos, la información es cualitativa y por lo tanto existen distintas formas de representarla. A la fecha, no existe un formato estándar para registrar la información. Esto dificulta la comparación de resultados entre distintos estudios a menos que hayan sido realizados por el mismo equipo de especialistas. En base a esta premisa existe la necesidad de una herramienta para almacenar los datos en estudios de vida pública, que sea de fácil acceso y que permita visualizarlos de manera en un formato estándar. Con este tipo de herramienta sería más sencillo analizar el comportamiento de las personas en distintos lugares a la vez y, en base a la información, plantear modelos que simulen el comportamiento peatonal, realizar estudios empíricos de flujos peatonales, entre otras aplicaciones.



CAPÍTULO 3: METODOLOGÍA

De acuerdo con la definición propuesta por el Project Management Institute, un proyecto es un esfuerzo temporal llevado a cabo con la finalidad de crear un nuevo producto, servicio o resultado (Project Management Institute, 2013). Para realizarlo de manera exitosa es recomendable definir una metodología que brinde lineamientos y protocolos definidos que faciliten la etapa de desarrollo. Esta metodología debe plantearse de acuerdo a las limitaciones y alcances del proyecto. El presente capítulo define la metodología utilizada en el presente proyecto.

Como se mencionó en el primer capítulo, la limitación más importante del presente trabajo es que el proyecto será desarrollado por una sola persona que no ha seguido una especialización en ingeniería de software ni ramas afines. Es por ello que la etapa de desarrollo no se acoplará exactamente a una metodología formalmente definida, sino que tratará de simplificarse para facilitar el desarrollo del programa. A continuación, se presentan las definiciones de la metodología propuesta, así como las herramientas que serán empleadas a lo largo del proyecto.

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad". El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (Jacobson, 1999).

Dentro de la ingeniería de software, a un nivel micro, existen actividades que definen las etapas necesarias para desarrollar un programa. Por otro lado, a un nivel macro, existen distintos modelos los cuales proponen enfoques sobre cómo organizar las etapas a lo largo del ciclo de vida del proyecto.

3.1. Etapas de desarrollo de software

Se identifican cinco etapas para el desarrollo de un software: estudio de requerimientos y análisis, diseño, implementación, periodo de pruebas y documentación. Estas etapas son secuenciales, a excepción de la documentación que se desarrolla durante todo el proyecto, aunque muchas veces sea necesario realizar modificaciones en etapas previas. A continuación, se presenta a detalla las etapas mencionadas.

3.1.1. Análisis de requerimientos

El desarrollo de un software o una herramienta informática responde a la necesidad planteada por un cliente, no necesariamente el usuario final de la aplicación pero que es quien define los requisitos que el programa debe cumplir, así como las funcionalidades que este debe tener. Comúnmente el cliente tiene una visión bastante abstracta del producto final y carece de claridad al momento de definir las funcionalidades que el programa debería cumplir. Si éste agregara nuevos requerimientos en etapas posteriores del ciclo de vida del proyecto se podría generar rediseños y procesos que dificultarían el desarrollo de la aplicación.

Por ello es importante definir y registrar de la manera más clara y concisa posible a que necesidades va a responder el software que se desea desarrollar para que, en base a estos requerimientos, se diseñe y planifique las siguientes etapas.

Usualmente estos requerimientos son definidos por el cliente/usuario, en el presente proyecto estos requerimientos son definidos por el alumno y el asesor como especialista del tema. Se describirá el comportamiento esperado, la forma en la que el programa interactuará con el usuario y el producto final esperado.

3.1.2. Diseño

Esta etapa consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la etapa anterior. Para ello es necesario conocer a profundidad las herramientas y funciones nativas del lenguaje de programación escogido para definir cómo éstas se utilizarán para atender a los requerimientos previamente definidos. En caso de no contar con las funciones requeridas, éstas deberán ser implementadas.

En primer lugar, se diseña la interfaz gráfica de usuario. Es a través de ésta que el usuario interactúa con el programa. Por ello, se desea que la interfaz sea intuitiva de manera que la secuencia de los pasos a seguir para ejecutar correctamente el programa sea sencilla. En la interfaz se distribuyen los elementos como botones, sliders, cuadros de texto, entre otros. Cuando el usuario interactúa con algún elemento se genera un evento y se ejecuta la función asociada a dicho evento.

Luego se procede a diseñar las funciones que serán ejecutadas al producirse un evento. Estas funciones pueden solicitar datos de entrada al usuario, almacenar información en variables o archivos o mostrar información en pantalla. Cabe señalar que, en este proyecto en particular, la información ingresada no recibe ningún tratamiento dentro del programa. Por ello, no es necesario implementar funciones de cálculo. Bastará con procedimientos que organicen, almacenen y muestren la información.

Finalmente se tiene el diseño de los directorios o carpetas en donde se almacenará el programa y los archivos que se generen. Estos archivos pueden ser los códigos del programa, archivos auxiliares que se utiliza para el funcionamiento o los productos que se obtengan del programa.

Esta es la etapa más importante del proyecto ya que un mal diseño conllevará a repetir procedimientos en las etapas posteriores. Es un símil con la etapa de diseño y planificación en un proyecto de construcción.

3.1.3. Implementación

En esta etapa se traduce el diseño antes definido a código y se ejecutan pruebas para verificar que el programa funciona correctamente. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados “tangibles”. No necesariamente es la etapa más larga ni la más compleja, aunque una especificación o diseño incompleto u ambiguo pueden exigir que tareas propias de las etapas anteriores se tengan que realizarse en esta. Esta etapa se dividirá en la implementación de las interfaces gráficas y la implementación de las funciones.

Para esta etapa se seguirán las denominadas “buenas prácticas de programación” (Johnson, 2002). Estas prácticas representan estándares para el desarrollo de un código que pueda ser entendido por cualquier programador y que facilite la posterior etapa de mantenimiento y un eventual proyecto de ampliación del programa. Por ejemplo, una de las prácticas recomendadas es el uso de nombres de variables que describan para qué sirve cada dato y así evitar nombres genéricos como “x” o “N”. Otra buena práctica es la inclusión de comentarios a lo largo del código describiendo las instrucciones que se llevan a cabo.

Todas las funciones deben mantener las relaciones previamente definidas en la etapa de diseño. Esto quiere decir que las variables y estructuras de datos utilizadas deben seguir un mismo formato. Es importante que el formato sea consistente para que los datos sean actualizados y compartidos entre las distintas interfaces.

Las pruebas realizadas estarán orientadas a verificar que el programa ejecute correctamente todas las funciones desencadenadas tras cada evento de la interfaz. De esta manera se pueden identificar errores, también llamados bugs, en el código y corregirlos antes de pasar a las pruebas finales.

3.1.4. Prueba y mantenimiento

Finalizado el punto anterior, se podría considerar que el programa está concluido; no obstante, es necesario contemplar un periodo de prueba. Este periodo consiste en comprobar que el software responda y realice correctamente las tareas indicadas en las especificaciones al ser utilizado por usuarios ajenos al proyecto. Para ello se distribuirá una copia a varios colaboradores para que la utilicen y reporten los errores que se presenten. El objetivo de esta etapa es identificar y corregir los errores y limitaciones que no hayan sido contemplados ni identificados durante la implementación del proyecto pero que requieren ser atendidos para cumplir con los requisitos inicialmente planteados. La duración del periodo de prueba puede extenderse dependiendo de los errores que se presenten.

Una vez terminada la etapa de prueba se da por concluido el proyecto. Cabe señalar que, es factible, aún existan errores y observaciones, pero estas escapan al alcance del proyecto.

3.1.5. Documentación

Finalmente, es necesario la redacción de un manual de usuario que detalle el correcto uso del programa; y un manual técnico con el objetivo de llevar a cabo un mantenimiento futuro, así como, ampliaciones al sistema. Las tareas de esta etapa se inician ya en la primera fase, tanto como parte del manuscrito de la tesis y como requerimiento de todo proyecto de software, pero sólo finalizan una vez terminadas las pruebas.

En la documentación se registra las etapas del proyecto a detalle, a manera de memoria descriptiva.

En los proyectos de ingeniería de software se reconoce una sexta etapa denominada mantenimiento. En esta etapa se realiza un mantenimiento correctivo, entendiéndose mantenimiento correctivo como resolver errores una vez puesto en marcha el programa, y un mantenimiento evolutivo definido como mejorar las funcionalidades y/o dar respuesta a nuevos requisitos. Cabe señalar que, esta etapa escapa al alcance del proyecto por lo que no será considerada.

Todas las etapas han sido planteadas en base a la investigación previa sobre Ingeniería de Software por lo que representan un punto de vista general sobre las etapas que todo proyecto de software debe considerar.

3.2. Modelos de desarrollo de software

Los modelos son representaciones abstractas del proceso de desarrollo de software. Proporcionan una metodología para llevar a cabo las distintas etapas del ciclo de vida del proyecto de acuerdo a las necesidades del mismo. A continuación, se expondrán tres de estos modelos, sus ventajas y desventajas y los casos en los que se recomienda aplicarlos.

3.2.1. Modelo en cascada

Es el modelo más simple y clásico, propuesto por Winston Royce en la década de los setenta. Esta metodología propone ordenar las etapas de la ingeniería de software de manera secuencial de manera que para iniciar una determinada etapa se debe concluir con la etapa precedente. En la Figura 3.1 se muestra un esquema de este modelo.

Las etapas que plantea este modelo son las mismas descritas en la sección 2.1. Una vez concluida cada etapa se pasa por un proceso de revisión para determinar si efectivamente la etapa ha sido concluida y se puede continuar con la siguiente. El objetivo de esta revisión es que una vez concluida una etapa, ésta ya no debería ser modificada o actualizada posteriormente.

Es en este punto donde radica la mayor desventaja de este modelo dado que a lo largo del ciclo de vida de un proyecto se presentan nuevos

requisitos y modificaciones en base a los resultados que se van obteniendo en las distintas pruebas. Estas modificaciones conllevan a un reproceso en gran parte de la cadena de desarrollo, lo que se traduce en pérdida y gasto de recursos. Por ello, esta metodología es recomendable en proyectos donde los requerimientos son claros desde un inicio y la posibilidad de presentarse modificaciones es muy baja.

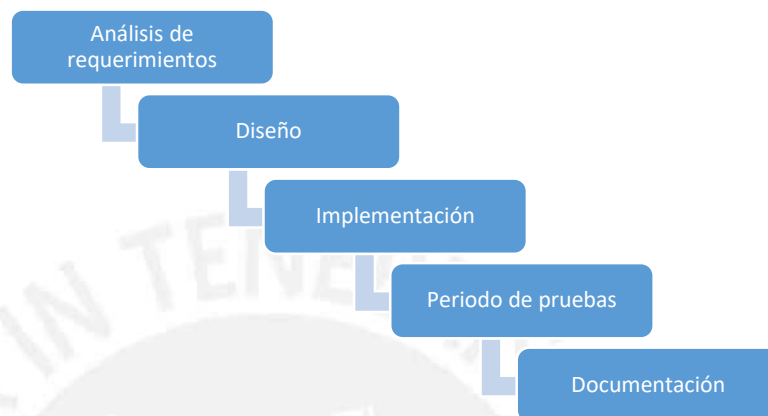


Figura 3.1. Esquema del modelo en cascada.
Fuente: Elaboración propia

3.2.2. Modelo prototipado

El modelo prototipado se basa en un paradigma evolutivo. La fortaleza de este modelo radica en que el producto inicia su desarrollo en base a objetivos generales y, a medida que avanza con el programa, se van refinando los detalles. En la Figura 3.2 se muestra un esquema del modelo en donde se puede apreciar su naturaleza cíclica. Como su nombre sugiere, el modelo se basa en el desarrollo de prototipos. Cada prototipo representa una evolución en el cual se añaden funciones y herramientas que, eventualmente, desembocarán en el programa final.

Se debe iniciar definiendo el núcleo del programa, es decir las funcionalidades y requisitos más importantes del proyecto. Luego se procede a un diseño rápido del cual se obtiene el primer prototipo. Este primer prototipo se presenta al cliente y/o usuarios y en base a sus observaciones se realizan modificaciones y correcciones o se procede a continuar con la implementación de nuevos prototipos.

Este modelo ayuda al desarrollador y al cliente a entender de mejor manera cuál será el resultado obtenido cuando los requisitos estén

satisfechos. Permite al cliente tener una visión más clara del producto que solicita e identificar aspectos que inicialmente no había contemplado.

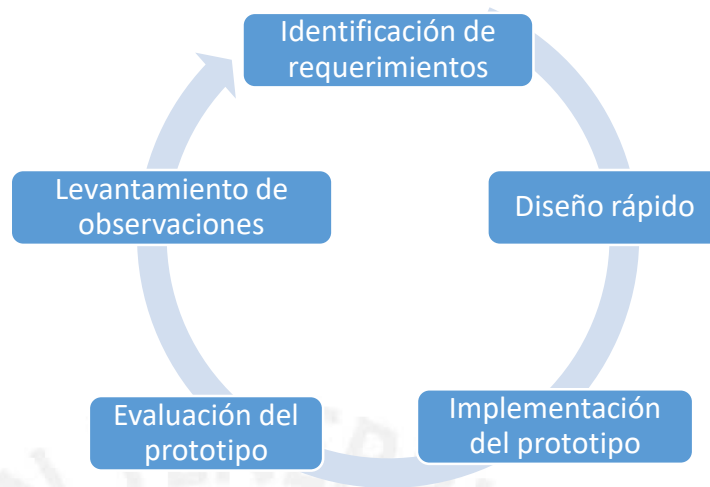


Figura 3.2. Esquema del modelo prototipado.
Fuente: Elaboración propia

3.2.3. Modelo incremental

El modelo incremental, también llamado creciente, fue planteado por Harlan Mills en la década de los ochenta con el objetivo de dar una respuesta a las deficiencias del modelo de cascada. Este modelo propone llevar a cabo el proceso de ingeniería de software en base a secuencias lineales, de manera similar al modelo de cascada, pero bajo un enfoque iterativo propio del modelo prototipado. En cada una de estas iteraciones se llevan a cabo las etapas de análisis de requerimientos, diseño, codificación y prueba; que producen incrementos en el software final. La figura 3.3 muestra un esquema de cómo funciona este modelo.

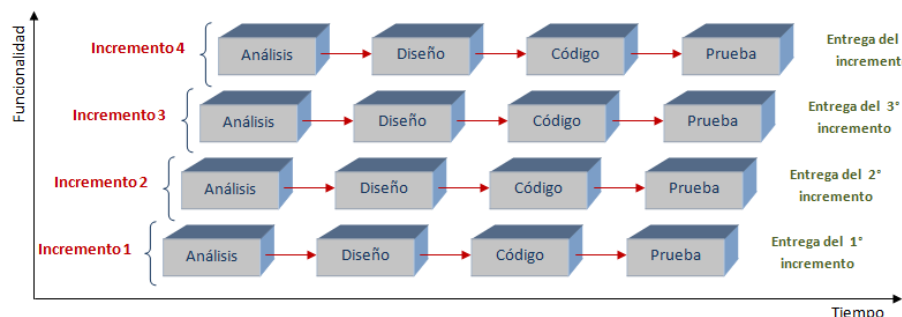


Figura 3.3. Esquema del modelo incremental.
Fuente: procesossoftware.wikispaces.com/Modelo+Incremental
Visitado 15/04/16

Este modelo es empleado en proyectos que pueden dividirse en subproyectos. En cada incremento se desarrolla un subproyecto por completo. Lo recomendable es definir un orden en los incrementos que permita identificar nuevos requerimientos para etapas posteriores. Se debe iniciar con aquel que se encuentra mejor definido y cuyo desarrollo permitirá tanto al cliente como al desarrollador comprender de mejor manera los alcances y limitaciones del producto.

De acuerdo a los requisitos del proyecto y los modelos expuestos, se decidió seguir un modelo incremental para aprovechar que el proyecto consiste en distintas herramientas independientes para procesar datos. En cada incremento se desarrollará una herramienta. Esto permitirá que el desarrollador se familiarice con el lenguaje de programación e identifique las limitaciones que se presenten en el desarrollo de herramientas más complejas.

El proyecto será dividido en siete incrementos. El primer incremento será la pantalla principal del programa desde la cual se administrará los proyectos. Los seis incrementos restantes serán las interfaces de las herramientas implementadas. La primera herramienta implementada será la de trazado/tracing. Luego se procederá a desarrollar la herramienta de seguimiento/tracking y mapeo/mapping. Posteriormente se implementará las herramientas de conteo/counting y tes de velocidad/Speed Test. Finalmente, el último incremento será la herramienta de diario/diary.

Para cada incremento se seguirá las etapas de análisis, diseño e implementación. Luego de culminar con todas las herramientas se procederá a realizar la etapa de pruebas. Durante el periodo de pruebas se desarrollará la guía de usuario y el manual técnico, elementos que pertenecen a la etapa de documentación.

CAPÍTULO 4: ANÁLISIS

La etapa de análisis es la primera etapa del proyecto. En ésta, se identifica el objetivo del programa en base a las necesidades de los usuarios finales. Para ello se utilizan dos conceptos propios de la ingeniería de software: requerimientos y casos de uso del programa.

Por un lado, los requerimientos son los requisitos que el cliente o usuarios solicitan para satisfacer sus necesidades por las que solicitan el desarrollo del programa. Los requerimientos se dividen en requerimientos funcionales y no funcionales. Los requerimientos funcionales están asociados a los servicios que provee el programa y la forma en la que este debe y no debe actuar ante las acciones del usuario. Los requerimientos no funcionales detallan características que no están asociadas a las funcionalidades del programa pero que aseguran que éste se desempeñe adecuadamente. Por ejemplo, definen restricciones del sistema como el tiempo de ejecución o la capacidad de almacenamiento. Por otro lado, un caso de uso es una descripción de los pasos necesario para completar un proceso. Dentro del contexto de la ingeniería de software, un caso de uso detalla las interacciones entre los usuarios y el sistema. Todos los requerimientos y casos de uso deben ser consistentes, es decir que éstos no deben contradecirse. Como se mencionó en el capítulo anterior, una correcta y completa identificación de los requerimientos y casos de uso facilita y agiliza las posteriores etapas del proyecto.

Los requerimientos del programa y los casos de uso se identificaron luego de analizar, en conjunto con el asesor y en base a la publicación *How to Study Public Life* (Gehl & Svarre, 2013), las principales herramientas utilizadas para la recolección de datos en los estudios de vida pública. En total se identificaron seis herramientas, las cuales serán implementadas en el programa. Cada herramienta posee requerimientos y casos de uso particulares. Por ello se decidió desarrollar una interfaz para cada herramienta. Además de las seis interfaces, se vio necesaria la creación de una interfaz principal desde la cual se ejecuta cada herramienta. De esta manera, siguiendo el modelo incremental elegido en el capítulo anterior, cada interfaz corresponderá a un incremento del programa.

El presente capítulo se divide en tres secciones. Primero se presenta los requerimientos funcionales de cada interfaz, luego se detalla los casos de uso identificados en cada incremento y finalmente, se enlista los requisitos no funcionales del sistema global identificados.

4.1. Requerimientos Funcionales

Para la identificación de los requisitos funcionales se analiza los objetivos del programa. Como se definió en el Capítulo 1, el principal objetivo del proyecto es proponer una alternativa para estandarizar los datos recolectados en estudios de vida pública para su posterior análisis. Por lo tanto, el requerimiento funcional más importante es la administración eficiente de los datos ingresados en el programa y, en segundo lugar, la capacidad de visualizar la información procesada de manera óptima. Los demás requisitos funcionales definidos buscan simplificar el uso del programa para que sea usado por cualquier usuario. A continuación, se detallan todos los requerimientos funcionales identificados para cada interfaz del programa.

4.1.1. Pantalla Principal

El objetivo de esta interfaz es administrar las herramientas implementadas. El programa trabaja en base a proyectos. Cada proyecto puede almacenar archivos generados por las distintas herramientas disponibles. Por lo tanto, esta interfaz debe permitir crear nuevos proyectos. El nombre asignado a cada proyecto deberá ser único con el objetivo de poder diferenciarlo de los demás proyectos.

Se desea que un proyecto sea modificado en distintas sesiones. Por ello, el programa debe ser capaz de guardar un proyecto incluyendo todos los datos ingresados. Para complementar esta funcionalidad, el programa debe ser capaz de abrir un proyecto antes guardado para continuar con la sesión.

Otro requerimiento identificado solicita que el programa sea capaz de compartir proyectos entre distintos equipos. De esta manera la información puede ser compartida y se puede añadir nuevos registros de datos desde distintas computadoras. Para ello, el programa necesita implementar funcionalidades para importar y exportar proyectos de manera sencilla.

Una vez creado un proyecto, desde esta interfaz se debe ingresar el lugar de estudio y el nombre del usuario. Además, desde esta pantalla se llama a las demás herramientas implementadas. La interfaz principal debe ser capaz de crear nuevos registros de cada herramienta y de abrir registros antes guardados. Mientras una de las herramientas esté habilitada la

pantalla principal debe estar inhabilitada y no visible. Cuando el usuario cierre la ventana de la herramienta la pantalla principal volverá a habilitarse.

4.1.2. Herramienta *Tracing*

Esta interfaz permite ingresar la información recolectada a través de la herramienta *Tracing*. Como se mencionó en el Capítulo 2, esta herramienta permite identificar las rutas, o líneas de deseo, que siguen los usuarios a través del espacio público analizado. Cada ruta es registrada como una línea sobre un plano del espacio analizado.

En primer lugar, es necesario que esta interfaz cargue información del proyecto como el nombre del mismo, el lugar de estudio y el nombre del usuario. Estos datos son ingresados en la interfaz principal del proyecto. Además de lo obtenido en la pantalla principal, otros datos importantes también deben ser definidos. El usuario puede registrar la fecha y hora del trabajo de campo. También es posible definir las condiciones climáticas, factor importante del comportamiento de las personas. Finalmente, debe existir un campo donde el usuario pueda ingresar Información adicional que considere necesaria.

En segundo lugar, es necesario definir el espacio público analizado. Para ello se deberá ingresar un croquis visto en planta sobre el cual se registrarán las líneas de deseo. Para facilitar el ingreso del croquis el programa debe aceptar archivos en formato JPG del mapa.

Los trazos deben ser ingresados uno a la vez. La forma más sencilla e intuitiva de ingresar los trazos es dibujarlos con el mouse por lo que esta funcionalidad debe ser implementada. También, cada trazo debe ser identificable entre los trazos existente, por ello cada uno debe contar con un nombre único. El programa permitirá personalizar el trazo asignándole un color y datos asociados a la persona observada. Los datos que pueden ser ingresados son el sexo, edad y la presencia de alguna discapacidad.

Los trazos deben poder ser modificados luego de haber sido creados, de esta manera el usuario podrá actualizar la información asociada o redibujar la trayectoria. Para agilizar el ingreso de datos también se debe implementar la función de duplicar un trazo. De esta manera el nuevo elemento creado mantendrá la configuración del trazo original.

Finalmente, el programa debe permitir eliminar un trazo existente cuando el usuario lo requiera.

Con respecto a la visualización de los datos ingresados, el programa debe poder filtrar los trazos mostrados. Los filtros serán aplicados de acuerdo a la información asociada a cada trazo. Con esta funcionalidad el usuario podría, por ejemplo, visualizar solo las rutas de personas de sexo femenino que tengan más de 18 años y no sufran de ninguna discapacidad. Finalmente, el programa debe ser capaz de generar un archivo PDF con los trazos mostrados sobre el croquis. De esta manera se puede compartir la información sin necesidad de que el receptor cuente con el programa.

4.1.3. Herramienta *Mapping*

Desde esta ventana, el usuario ingresa los datos obtenidos al mapear las actividades en el espacio de interés. A través de esta herramienta se clasifica a las personas según la actividad que se encuentren realizando. Además, esta información puede ser filtrada con base en criterios como sexo, edad y la presencia de discapacidades. Similar a la herramienta anterior, es necesario que la interfaz muestra la información general del proyecto. Los datos de hora, fecha y condiciones climáticas pueden ser modificadas. Además, se brinda un espacio en donde el usuario puede registrar una nota de texto.

El primer paso para trabajar con la herramienta es cargar un croquis del espacio analizado. El programa debe ser capaz de cargar cualquier archivo en formato JPG y mostrarlo en pantalla de manera instantánea. Luego, se debe brindar al usuario la flexibilidad para definir las actividades que puede ingresar en el croquis. Cada actividad será identificada con un nombre único y podrá ser eliminada según el usuario lo requiera. Para ubicar los puntos donde se desarrolla cada acción, el medio más sencillo es a través del mouse. Cada actividad debe tener la capacidad de ser personalizable de manera que se pueda definir el color, tipo y tamaño de la marca sobre el plano y la información que representa con respecto a edad, sexo y presencia de discapacidades. Además, el programa debe ser capaz de editar los puntos ingresados ya sea reubicándolos o eliminándolos.

En relación a la visualización, al igual que las demás herramientas, se necesita imprimir los datos en un archivo en formato PDF que resuma la información ingresada. Las actividades mostradas deben poder ser filtradas según los criterios de edad, sexo y presencia de discapacidades. En el documento se debe presentar, además de los datos globales del proyecto, el croquis con las actividades filtradas y una leyenda que permita identificarlas.

4.1.4. Herramienta *Tracking*

Esta interfaz tiene el objetivo de presentar los registros obtenidos mediante la herramienta de seguimiento. Como se mencionó en la revisión de literatura, cuando el espacio analizado resulta extenso para analizar el movimiento de un conjunto de personas al mismo tiempo, resulta más eficiente registrar las actividades de una persona en particular.

Similar a la herramienta Tracing, el usuario necesita identificar el espacio analizado a través de un plano; no obstante, en esta ocasión solo es necesario ingresar una ruta correspondiente al desplazamiento de la persona analizada. El medio para hacerlo debe ser igual al utilizado en la herramienta tracing, a través del movimiento del cursor.

Los datos recolectados durante el desplazamiento deben poder ser añadidos como registros. Cada registro necesita estar asociado a un punto que es ubicado sobre el croquis y a una nota de texto editable por el usuario. Dichos registros deberán poder ser modificados o eliminados según el usuario lo requiera. Cada registro será identificado con un número para diferenciarlos.

Por último, en reporte en formato PDF generado se deberá presentar la información ingresada. Por un lado, presentar el croquis donde se muestra tanto la trayectoria ingresada como los puntos asociados a los registros. En el plano debe ser posible identificar cada punto con su número correspondiente. Por otro lado, se debe mostrar las notas de texto asociadas los puntos presentes en el plano.

4.1.5. Herramienta *Counting*

Esta interfaz está asociada a la herramienta conteo. A través de ésta se permite ingresar, a través de valores numéricos, los flujos peatonales a

través de una sección analizada. Las cantidades pueden ser medidas en distintos intervalos de tiempo. La interfaz, similar a las demás deberá mostrar la información general del proyecto.

Como los datos ingresados representan el flujo peatonal a través de una sección, el usuario necesita identificar dicha sección en un croquis. La sección única y deberá ser ingresada a través del mouse. Como la cantidad de personas depende de la hora en la que se realice la medición, el usuario necesita ingresar la hora de inicio del registro. Además, como es costumbre en este tipo de mediciones, se divide el periodo de análisis en intervalos de tiempo homogéneos. El usuario debe ser capaz de definir la duración y cantidad de los periodos. Finalmente, para el ingreso de datos se debe permitir al usuario ingresar los datos directamente en el programa. En caso la cantidad de datos sea grande, el programa también necesita ser capaz de importar datos desde una fuente externa, siendo la más usual una hoja de Excel. Cada dato ingresado podrá ser identificado con las características de sexo, edad y presencia de discapacidades para su mejor interpretación.

Con base en los valores ingresados, el programa debe mostrar un histograma con los flujos en cada periodo. Además, según los valores también se debe mostrar datos estadísticos simples como valores máximos, valores mínimos, promedio, desviación estándar, entre otros. Esta información debe poder ser filtrada siguiendo los mismos criterios que el resto de herramientas. Además, también debe ser posible seleccionar qué periodos son de interés.

En el reporte PDF generado, se debe mostrar el croquis en donde se muestre la sección analizada. También es importante presentar el histograma obtenido y el cuadro con los parámetros estadísticos calculados.

4.1.6. Herramienta *Speed Test*

A través de esta ventana se ingresará los datos recolectados en los recorridos de prueba que son mencionados en el Capítulo 2. De manera similar a la herramienta anterior, se ingresan valores numéricos. La diferencia radica que en esta herramienta no registra cantidades, sino velocidades promedio.

En el espacio analizado, que es ingresado como una imagen, se identifica las secciones que delimitan el punto de inicio y de fin del recorrido analizado. Estas secciones deben ser ingresadas como trazos sobre el croquis.

Para el registro de los valores numéricos se permitirá el ingreso de los tiempos que toma el recorrido, que son los datos registrados en campo. De esta manera, la velocidad será calculada por el programa según la distancia definida por el usuario. Similar a la herramienta counting, los datos podrán ser ingresados directamente o importados desde una hoja de Excel. De igual manera, la información será mostrada en un histograma de las velocidades calculadas. El programa debe permitir seleccionar las unidades que se desean utilizar: ya sea metros por segundo o kilómetros por hora. Algunos parámetros estadísticos serán mostrados en una tabla. Estos datos serán actualizados según los criterios que el usuario defina.

El formato del PDF generado por esta herramienta será igual a la herramienta anterior. La diferencia radica en la información mostrada ya que en el croquis se muestra dos secciones en lugar de una.

4.1.7. Herramienta *Diary*

Esta última interfaz tiene como objetivo servir de diario para registrar detalles que el usuario haya identificado en el trabajo de campo. A diferencia de las demás herramientas, en ésta no es necesario habilitar un botón para añadir notas ni un croquis ya que cada registro permite ingresar ambos elementos.

El programa debe permitir ingresar múltiples registros. En cada registro, el usuario podrá identificar la hora y fecha asociada al registro. Además, podrá redactar una nota de texto y añadir una imagen de formato JPG.

Con respecto el documento en formato PDF generado, éste deberá incluir todos los registros ingresados. Cada registro estará compuesto del texto y la fotografía. Cabe señalar que no debe ser necesario ingresar ambos elementos, es posible ingresar texto y no imagen o viceversa.

4.2. Casos de uso

Como se mencionó anteriormente, los casos de uso detallan los pasos necesarios para ejecutar un proceso dentro del programa. Un caso de uso puede agrupar distintas funcionalidades del programa. Se han identificado casos de uso para cada una de las interfaces desarrolladas en base a los requerimientos funcionales identificados en la sección anterior.

Cabe señalar que, dentro de las definiciones en el contexto de los casos de uso, el ente que desencadena o inicio el proceso se denomina “actor”. Los actores pueden de diferentes de acuerdo a las acciones que realizan dentro del programa. Sin embargo, en el presente proyecto solo existe un tipo de actor que es el usuario final del sistema. Por ello, no será necesario presentar esta información en cada caso de uso

Cada caso de uso será presentado en una tabla. Primero, se especifica el proceso analizado. También se detallan, si existiesen, las condiciones necesarias para iniciar con el proceso. Luego, se enlista los pasos que el usuario debe seguir para ejecutar el proceso. Finalmente se muestran las postcondiciones, o resultados, que resultan luego de ejecutar el proceso. A continuación, se muestran los casos de uso identificados.

4.2.1. Pantalla principal

Dentro de la interfaz de la pantalla principal se analizan tres casos de uso. En primer lugar, el caso de uso para habilitar un proyecto, ya sea al crear uno nuevo o al abrir uno existente, se presenta en la Tabla 4.1. En segundo lugar, el caso de uso para almacenar el proyecto activo, el cual puede ser guardado en el equipo o exportado y compartido con otro usuario, se muestra en la Tabla 4.2. Finalmente, en la Tabla 4.3, se analiza el caso de uso para modificar la información del proyecto activo: modificar los datos generales del proyecto, crear un nuevo archivo usando las herramientas implementadas o abrir un archivo existente.

Tabla 4.1. Caso de uso Activar un proyecto.

Caso de uso: Activar un proyecto	
Descripción	El propósito de este caso de uso es habilitar un proyecto para poder añadir o modificar información dentro de él.
Precondición	El usuario debe abrir el programa.
Crear un proyecto	
<ol style="list-style-type: none"> 1. El usuario activa el botón Crear Proyecto. 2. El programa solicita ingresar un nombre para el nuevo proyecto. 3. El programa reconoce si existe un proyecto con el mismo nombre ingresado. En tal caso se da la opción al usuario de sobrescribir dicho proyecto o ingresar un nuevo nombre. 4. Se crea las carpetas del nuevo proyecto y se habilitan los campos para añadir información y crear registros con las herramientas. 	
Postcondición	El nuevo proyecto se encuentra activo en el programa.
Abrir un proyecto	
<ol style="list-style-type: none"> 1. EL usuario activa el botón Abrir Proyecto. 2. El sistema muestra una lista de los proyectos existentes. 3. El usuario selecciona uno de los proyectos existentes. 4. El programa carga la pantalla principal del proyecto seleccionado. 	
Postcondición	El nuevo abierto se encuentra activo en el programa.
Importar un proyecto	
<ol style="list-style-type: none"> 1. EL usuario activa el botón Importar Proyecto. 2. El sistema muestra una lista de los proyectos exportados en el equipo. 3. El usuario selecciona el archivo del proyecto que desea importar, no necesariamente dentro de la carpeta que el programa muestra por defecto. 4. La información del proyecto es añadida a la carpeta de proyectos dentro del equipo. 	
Postcondición	El proyecto importado se encuentra dentro de los proyectos existentes en el equipo.

Fuente: Elaboración propia.

Tabla 4.2. Caso de uso Salvar un proyecto.

Caso de uso: Salvar un proyecto	
Descripción	El propósito de este caso de uso es almacenar la información del proyecto activo.
Precondición	Un proyecto debe estar activo en el programa.
Guardar un proyecto	
<ol style="list-style-type: none"> 1. El usuario activa el botón Guardar Proyecto. 2. El programa genera un archivo dentro de la carpeta del proyecto con la información necesaria para restablecer la ventana actual. 	
Postcondición	Se ha creado un archivo que puede ser abierto en otra sesión.
Exportar un proyecto	
<ol style="list-style-type: none"> 1. EL usuario activa el botón Exportar Proyecto. 2. El programa genera un archivo en formato ZIP dentro de la carpeta "Exportados". El archivo contiene toda la información actual del proyecto. 	
Postcondición	Se ha creado un archivo que puede ser importado por otro usuario.

Fuente: Elaboración propia.

Tabla 4.3. Caso de uso Modificar información del proyecto.

Caso de uso: Modificar información del proyecto	
Descripción	El propósito de este caso de uso es modificar la información del proyecto activo
Precondición	Un proyecto debe estar activo en el programa
Modificar datos generales	
1. El usuario activa modifica el texto de los campos "Lugar de estudio" o "Nombre de Usuario". 2. El sistema muestra el nuevo texto ingresado por el usuario. 3. Se actualiza los datos ingresados por el usuario en las variables del programa	
Postcondición	Se ha actualizado la información.
Crear nuevo registro	
1. El usuario selecciona la opción "Nuevo Archivo" en uno de los menús desplegados de las herramientas disponibles. 2. El usuario activa el botón abrir. 3. El programa solicita un nombre para el nuevo archivo. 4. Si el nombre ingresado por el usuario se encuentra en uso se pregunta si desea sobrescribir el archivo o ingresar un nuevo nombre. 5. La ventana principal es cerrada y se abre la interfaz de la herramienta seleccionada.	
Postcondición	Se habilita la interfaz de la herramienta seleccionada.
Abrir registro existente	
1. El usuario selecciona el nombre del archivo que desea abrir en uno de los menús desplegados de las herramientas disponibles. 2. El usuario activa el botón abrir. 5. La ventana principal es cerrada y se abre la interfaz de la herramienta seleccionada con la información del registro seleccionado.	
Postcondición	Se habilita la interfaz de la herramienta seleccionada
Actualizar archivos del proyecto	
1. El usuario activa el botón "Actualizar archivos". 2. El programa actualiza los menús desplegados de todas las herramientas y muestra los nombres de los archivos que existen dentro del proyecto. 3. Se habilita todos los elementos de la venta principal.	
Postcondición	Los menús de las herramientas se actualizan y se habilita los elementos de la ventana

Fuente: Elaboración propia.

4.2.2. Herramienta *Tracing*

En la interfaz de la herramienta *Tracing*, se analiza cuatro casos de uso. En primer lugar, en la Tabla 4.4 se muestra el caso de uso para modificar la información general del archivo: las condiciones climáticas, la fecha y hora asociada, el registro de notas de texto y la selección de la imagen de croquis del espacio analizado. Luego, en la Tabla 4.5, se analiza el caso de uso para crear un nuevo trazo. En éste se detalla los procesos para ingresar y configurar un nuevo trazo. Seguidamente, se analiza el caso de uso para administrar los trazos existentes en la Tabla 4.6 en donde se duplica, modifica o elimina un trazo. Finalmente, se presenta el

caso de uso para generar reportes en formato PDF y aplicar filtros para los trazos visualizados en la Tabla 4.7.

Tabla 4.4. Caso de uso Modificar información del registro.

Caso de uso: Modificar información del registro	
Descripción	El propósito de este caso de uso es modificar la información general del registro activo.
Precondición	El archivo debe estar activo. Puede ser un archivo nuevo o uno abierto.
Ingresar clima, fecha y hora.	
<ol style="list-style-type: none"> 1. El usuario modifica el texto en el campo "Clima". 2. El sistema muestra el nuevo texto ingresado por el usuario. 3. El usuario modifica los valores de fecha y hora. 4. Se actualiza los datos ingresados en las variables del programa 	
Postcondición	Se ha actualizado la información.
Añadir notas de texto.	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Ingresar Notas". 2. El programa muestra una ventana en donde el usuario puede ingresar texto. 3. Si el usuario ha ingresado texto previamente se muestra el texto antes ingresado. 	
Postcondición	Las notas son guardadas dentro de las variables del programa.
Importar croquis	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Importar Imagen". 2. El programa muestra la lista de imágenes, en formato JPG, en la carpeta "Imágenes" dentro del proyecto. 3. El usuario selecciona la imagen que desea usar. No necesariamente debe estar dentro de la carpeta mostrada por defecto. 4. Si el archivo no se encuentra dentro de la carpeta de imágenes, el programa crea una copia del archivo en esta carpeta. 5. Se muestra la imagen cargada en el elemento Axes de la interfaz. 6. Se habilita los paneles Trazos y Menú. 	
Postcondición	Se habilita los paneles Trazos y Menú.

Fuente: Elaboración propia.

Tabla 4.5. Caso de uso Crear un trazo.

Caso de uso: Crear un trazo	
Descripción	El propósito de este caso de uso es crear un nuevo trazo.
Precondición	La imagen del croquis debe estar cargada.
Crear un nuevo trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Crear Trazo". 2. El sistema solicita ingresar un nombre para el nuevo trazo. Por defecto el programa sugiere el nombre "Trazo x" donde x es el menor número que no esté utilizado. 3. Si el nombre elegido ya existe se da la opción al usuario de sobrescribir el trazo o ingresar un nuevo nombre. 4. Se habilita el panel Menú de Trazo y se muestra la información ingresada. Todos los trazos visibles se ocultan. 5. Se deshabilita el panel Trazos y Menú. 	
Postcondición	Se habilita el panel Menú de Trazo
Dibujar trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar Trazo". 2. El sistema habilita el elemento Axes para dibujar un trazo. 3. Manteniendo presionado el botón izquierdo del mouse, se dibuja el trazo con el cursor sobre el croquis. 4. El sistema muestra el trazo dibujado. 	
Postcondición	Se muestra el trazo dibujado.
Cambiar de color el trazo	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los colores básicos en el menú desplegable de color. 2. El usuario modifica las componentes rojo o verde o azul del color activo. 3. El sistema modifica el color del trazo, si ya estuviese dibujado. 4. El color de fondo del menú desplegable de color se actualiza al color seleccionado. 	
Postcondición	Se actualiza el color del trazo
Asignar datos al trazo	
<ol style="list-style-type: none"> 1. A través de menús desplegables, el usuario asigna datos de edad, sexo y discapacidad al trazo activo. 2. El sistema muestra la selección en el menú desplegable respectivo. 3. Se actualiza las variables del programa. 	
Postcondición	Se actualiza la información modificada.
Guardar trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Guardar". 2. El programa deshabilita el panel Menú de Trazo y habilita los paneles Trazo y Menú. 3. Se grafica todos los trazos habilitados sobre el croquis. 	
Postcondición	Se añade el trazo ingresado.

Fuente: Elaboración propia.

Tabla 4.6. Caso de uso Administrar trazos.

Caso de uso: Administrar trazos	
Descripción	El propósito de este caso de uso es administrar y manipular los trazos existentes.
Precondición	Seleccionar un trazo del menú desplegable en el panel Trazos
Modificar un trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Modificar trazo". 2. El programa habilita el panel Menú de trazo y carga la información almacenada del trazo seleccionado. 3. Se dibuja el trazo seleccionado sobre el croquis. 	
Postcondición	Se habilita el trazo seleccionado para modificar sus datos
Duplicar un trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Duplicar Trazo". 2. El sistema solicita un nombre para el nuevo trazo. Por defecto se sugiere el nombre del trazo original seguido de "2". 3. Si el nombre elegido ya existe se da la opción al usuario de sobrescribir el trazo o ingresar un nuevo nombre. El trazo sobrescrito no puede ser el trazo original. 4. Se habilita el panel Menú de Trazo con la información del trazo original, pero con el nuevo nombre ingresado. 	
Postcondición	Se habilita el panel Menú de Trazo con la información del duplicado copiado.
Eliminar un trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Eliminar Trazo". 2. El sistema solicita la confirmación del usuario para eliminar el trazo seleccionado. 3. Se elimina los datos del trazo y se quita del menú desplegable en el panel Trazos. 	
Postcondición	Se elimina el trazo seleccionado.

Fuente: Elaboración propia.

Tabla 4.7. Caso de uso Exportar datos.

Caso de uso: Exportar datos	
Descripción	El propósito de este caso de uso es exportar los datos ingresados en formato PDF.
Precondición	Ingresar el menos un trazo sobre el croquis,
Filtros de visualización	
<ol style="list-style-type: none"> 1. El usuario selecciona filtros de edad, sexo y/o discapacidad. 2. El programa muestra los trazos que cumplen los filtros seleccionado. 	
Postcondición	Se muestra en pantalla los trazos que cumplen filtros seleccionados.
Generar PDF	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Generar PDF". 2. El programa crea una nueva figura en donde se copia la información que se visualiza sobre el elemento Axes. 3. Se genera un archivo en formato PDF dentro de la carpeta "PDF", en la carpeta Tracing, dentro del directorio del proyecto. 4. El nombre del archivo generado es el nombre del registro actual seguido de un número natural, el primero que no esté en uso. 	
Postcondición	Se genera un archivo en formato PDF

Fuente: Elaboración propia.

4.2.3. Herramienta *Mapping*

En la interfaz de la herramienta Tracing, se analiza cuatro casos de uso. Los casos de uso presentados en las tablas 4.4 y 4.7 se repiten por lo que ya no serán presentadas en esta sección. La tabla 4.8 muestra el caso de uso para administrar las actividades, esto incluye la creación, edición y eliminación de las mismas. La tabla 4.9 resume el caso de uso de edición de una actividad en donde se muestran los procesos para añadir puntos, personalizarlos y eliminarlos.

Tabla 4.8. Caso de uso Administrar actividades.

Caso de uso: Administrar actividades	
Descripción	El propósito de este caso de uso es crear, administrar y eliminar las actividades utilizadas.
Precondición	La imagen del croquis debe estar cargada.
Crear una nueva actividad	
	<ol style="list-style-type: none"> 1. El usuario activa el botón "Añadir". 2. El sistema solicita ingresar un nombre para la actividad. Por defecto el programa sugiere el nombre "Actividad x" donde x es el menor número que no esté utilizado. 3. Si el nombre elegido ya existe se da la opción al usuario de sobrescribir el trazo o ingresar un nuevo nombre. 4. Se habilita el panel Actividad. Todos los puntos visibles se ocultan. 5. Se deshabilita el panel Actividades y Menú
Postcondición	Se habilita el panel Actividad
Dibujar puntos	
	<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar". 2. El sistema habilita el elemento Axes para dibujar puntos. 3. Al hacer clic sobre un punto en el croquis el programa lo dibuja. 4. Los puntos son añadidos uno tras otro. 5. Para dejar de dibujar el usuario activa el botón "Detener".
Postcondición	Se muestra los puntos dibujados.
Iniciar edición de una actividad	
	<ol style="list-style-type: none"> 1. El usuario selecciona en el menú desplegable una de las actividades creadas. 2. Se activa el botón editar. 3. Se habilita el panel Actividad. Todos los puntos visibles se ocultan. 4. Se deshabilita el panel Actividades y Menú
Postcondición	Se habilita el panel Actividad
Eliminar actividad	
	<ol style="list-style-type: none"> 1. El usuario activa el botón "Eliminar". 2. El sistema solicita la confirmación del usuario para eliminar la actividad seleccionada. 3. Se elimina los datos de la actividad y se retira del menú desplegable.
Postcondición	Se elimina la actividad seleccionada.

Fuente: Elaboración propia.

Tabla 4.9. Caso de uso Editar actividad.

Caso de uso: Editar actividad	
Descripción	El propósito de este caso de uso es editar una actividad existente.
Precondición	Seleccionar una actividad en el menú desplegable y activar el botón "Editar".
Dibujar puntos	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar". 2. El sistema habilita el elemento Axes para dibujar puntos. 3. Al hacer clic sobre un punto en el croquis el programa lo dibuja. 4. Los puntos son añadidos uno tras otro. 5. Para dejar de dibujar el usuario activa el botón "Detener". 	
Postcondición	Se muestra los puntos dibujados.
Personalizar los puntos	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los colores básicos en el menú desplegable de colores. 2. El usuario modifica las componentes rojo o verde o azul del color activo. 3. El usuario selecciona uno de los marcadores disponibles en el menú desplegable. 4. El usuario varía el tamaño del marcador usando una slider. 5. El sistema modifica el color de los puntos, el tamaño y tipo de marcadores, si ya estuviesen dibujados. 	
Postcondición	Se actualiza el color de los puntos.
Asignar datos a la actividad	
<ol style="list-style-type: none"> 1. A través de menús desplegables, el usuario asigna datos de edad, sexo y discapacidad al trazo activo. 2. El sistema muestra la selección en el menú desplegable respectivo. 3. Se actualiza las variables del programa. 	
Postcondición	Se actualiza la información modificada.
Eliminar puntos ingresados	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Iniciar" en el panel Eliminar. 2. El programa resalta el punto que se indique en el cuadro de texto del mismo panel. 3. El usuario puede seleccionar el punto que desee eliminar usando una slider. 4. Al activar el botón "Eliminar" se borra el punto resaltado. 5. Para terminar la edición se pulsa el botón "Terminar". 	
Postcondición	Se borran los puntos eliminados.
Guardar trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Guardar". 2. El programa deshabilita el panel Actividad y habilita los paneles Actividades y Menú. 3. Se grafica todos los puntos habilitados sobre el croquis. 	
Postcondición	Se guarda la actividad.

Fuente: Elaboración propia.

4.2.4. Herramienta *Tracking*

En esta interfaz se identificaron cuatro casos de uso. Las tablas 4.4 y 4.7 muestran dos de ellos. La tabla 4.10 muestra el caso de uso para el ingreso de la trayectoria analizada, dentro de éste se resume las actividades para dibujar la trayectoria e iniciar la edición de registros. En

la tabla 4.11 se presenta los pasos necesarios para agregar y editar registros.

Tabla 4.10. Caso de uso Definir trayectoria.

Caso de uso: Definir trayectoria	
Descripción	El propósito de este caso de uso es ingresar la trayectoria de la persona analizada.
Precondición	La imagen del croquis debe estar cargada.
Dibujar trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar Trayectoria". 2. El sistema habilita el elemento Axes para dibujar un trazo. 3. Manteniendo presionado el botón izquierdo del mouse, se dibuja el trazo con el cursor sobre el croquis. 4. El sistema muestra el trazo dibujado. 	
Postcondición	Se muestra la trayectoria dibujada.
Dibujar trazo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar Trayectoria". 2. El sistema habilita el elemento Axes para dibujar un trazo. 3. Manteniendo presionado el botón izquierdo del mouse, se dibuja el trazo con el cursor sobre el croquis. 4. El sistema muestra el trazo dibujado. 	
Postcondición	Se muestra la trayectoria dibujada.
Seleccionar color e información	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los colores básicos en el menú desplegable de color. 2. El usuario modifica las componentes rojo o verde o azul del color activo. 3. El sistema modifica el color del trazo, si ya estuviese dibujado. 4. El color de fondo del menú desplegable de color se actualiza al color seleccionado. 5. Los puntos de los registros también son actualizados. 6. A través de menús desplegables, el usuario asigna datos de edad, sexo y discapacidad al trazo activo. 7. Se actualiza las variables del programa. 	
Postcondición	Se actualiza el color e información ingresada
Iniciar edición de registros	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón "Editar Registros". 2. Se deshabilita el panel Menú de trazo y se habilita el panel Registros. 	
Postcondición	Se habilita la edición de los registros.

Fuente: Elaboración propia.

Tabla 4.11. Caso de uso Ingresar registros.

Caso de uso: Ingresar registros	
Descripción	El propósito de este caso de uso es añadir y modificar registros.
Precondición	Se debe habilitar el panel Registros.
Crear Registro	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Nuevo Registro". 2. Se crea el registro, este será identificado con el último número de la lista de registros existentes. 3. El número de registro se muestra en el cuadro de texto del panel. 	
Postcondición	Se crea el registro.
Ubicar punto	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Ubicar Punto". 2. El sistema habilita el elemento Axes para dibujar puntos. 3. Al hacer clic sobre un punto en el croquis el programa lo dibuja. 4. Se añade el número asociado al registro y se muestra en el gráfico. 	
Postcondición	Se dibuja el punto sobre el croquis.
Editar nota de texto	
<ol style="list-style-type: none"> 1. El usuario modifica el texto asociado al registro. 2. Se actualiza el texto y se almacena en el sistema. 	
Postcondición	Se modifica el texto del registro.
Eliminar Registro	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Eliminar Registro". 2. El sistema borra el registro sin solicitar la confirmación del usuario. 3. Los números asociados a los registros restantes son reasignados de manera que se mantenga la secuencia. 	
Postcondición	Se elimina el registro seleccionado.
Finalizar edición	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Terminar Edición". 2. Se deshabilita el panel Registros y se habilita los paneles Menú de Trazo y Menú. 3. Sobre el croquis se muestra los puntos de todos los registros ubicados. 	
Postcondición	Se actualiza el croquis.

Fuente: Elaboración propia.

4.2.5. Herramientas *Counting*

Similar a los casos anteriores, esta herramienta cuenta con cuatro casos, de estos dos se presentaron en las tablas 4.4 y 4.7. En la tabla 4.12 se muestra el caso de uso para definir los datos del registro en donde se dibuja la sección de interés y se definen los parámetros del intervalo de tiempo analizado. La tabla 4.13 resume los procedimientos que se deben seguir para ingresar los datos de manera individual o cargarlos desde una hoja de Excel.

Tabla 4.12. Caso de uso Definir datos del registro.

Caso de uso: Definir datos del registro	
Descripción	El propósito de este caso de uso es definir la sección de interés y los datos del periodo analizado.
Precondición	La imagen del croquis debe estar cargada.
Dibujar sección	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar Sección". 2. El sistema habilita el elemento Axes para dibujar un trazo. 3. Manteniendo presionado el botón izquierdo del mouse, se dibuja el trazo con el cursor sobre el croquis. 4. El sistema muestra el trazo dibujado. 	
Postcondición	Se muestra la sección dibujada.
Definir hora de inicio, duración y cantidad de periodos.	
<ol style="list-style-type: none"> 1. Utilizando los sliders en el panel se ingresa la hora a partir de la cual se inició la toma de datos. 2. En los campos habilitados se ingresa la duración de cada periodo, en minutos, y la cantidad de los mismos. 	
Postcondición	Se actualiza la información.

Fuente: Elaboración propia.

Tabla 4.13. Caso de uso Ingresar datos.

Caso de uso: Ingresar datos	
Descripción	El propósito de este caso de uso es ingresar los datos recolectados.
Precondición	La imagen del croquis debe estar cargada.
Abrir ventana para ingresar datos	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Cargar Datos". 2. Se abre una ventana en donde se muestra la tabla para ingresar los valores. 	
Postcondición	Se abre la ventana "Cargar datos".
Ingresar datos individuales	
<ol style="list-style-type: none"> 1. El usuario selecciona una de las celdas de la tabla. 2. Valores numéricos pueden ser ingresados a través del teclado. 3. Para las celdas de texto se despliega un menú en donde se puede seleccionar la opción deseada. 4. Para añadir una fila en la tabla se selecciona el botón "Añadir Fila". 	
Postcondición	El valor ingresado se muestra en la tabla.
Importar hoja de cálculo	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Importar Excel". 2. Se muestra una pantalla con las indicaciones para importar una hoja de Excel. 3. Se habilita una ventana para seleccionar el archivo desde cualquier carpeta en el equipo. 4. El programa carga la información y la muestra en la tabla. 5. Si alguna celda no corresponde al formato entonces el campo se deja vacío. 	
Postcondición	Se cargan los datos importados.
Terminar carga de datos	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Terminar". 2. La ventana "Cargar datos" se cierra y se actualiza la información mostrada en la pantalla de la interfaz. 3. Cuando se vuelva a activar la carga de datos se muestra los datos ingresados. 	
Postcondición	Se actualiza la información en pantalla.

Fuente: Elaboración propia.

4.2.6. Herramienta *Speed Test*

Esta herramienta es bastante similar a la anterior ya que en ambas se ingresan datos numéricos que son presentados en un histograma. Se identificaron cuatro casos de uso de los cuales dos se han venido repitiendo en las tablas 4.4 y 4.7. La tabla 4.14 muestra el caso de uso para definir las secciones de análisis y la tabla 4.15, para ingresar los datos.

Tabla 4.14. Caso de uso Definir secciones de análisis.

Caso de uso: Definir secciones de análisis	
Descripción	El propósito de este caso de uso es definir las secciones que marcan el inicio y fin de la distancia analizada.
Precondición	La imagen del croquis debe estar cargada.
Dibujar secciones	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Dibujar Secciones". 2. El sistema habilita el elemento Axes para dibujar dos trazos. 3. Manteniendo presionado el botón izquierdo del mouse, se dibuja el trazo con el cursor sobre el croquis. 4. El sistema muestra el trazo dibujado. 5. Se vuelve a habilitar los elementos de la ventana cuando se han dibujado los dos trazos. 	
Postcondición	Se muestra las secciones dibujadas.

Fuente: Elaboración propia.

Tabla 4.15. Caso de uso Ingresar datos.

Caso de uso: Ingresar datos	
Descripción	El propósito de este caso de uso es ingresar los datos recolectados.
Precondición	La imagen del croquis debe estar cargada.
Abrir ventana para ingresar datos	
<ol style="list-style-type: none"> 1. El usuario activa el botón "Cargar Datos". 2. Se abre una ventana en donde se muestra la tabla para ingresar los valores. 3. En la ventana debe ingresarse la distancia, en metros, del tramo analizado. 	
Postcondición	Se abre la ventana "Cargar datos".
Ingresar datos individuales	
<ol style="list-style-type: none"> 1. El usuario selecciona una de las celdas de la tabla. 2. Si la celda es para un valor numérico, éste puede ser ingresado a través del teclado. 3. Para las celdas de texto se despliega un menú en donde se puede seleccionar la opción deseada. 4. Para añadir una fila en la tabla se selecciona el botón "Añadir Fila". 	
Postcondición	El valor ingresado se muestra en la tabla.

Importar hoja de cálculo	
1. El usuario activa el botón "Importar Excel".	
2. Se muestra una pantalla con las indicaciones para importar una hoja de excel.	
3. Se habilita una ventana para seleccionar el archivo desde cualquier carpeta en el equipo.	
4. El programa carga la información y la muestra en la tabla. 5. Si alguna celda no corresponde al formato entonces el campo se deja vacío.	
Postcondición	Se cargan los datos importados.
Terminar carga de datos	
1. El usuario activa el botón "Terminar".	
2. La ventana "Cargar datos" se cierra y se actualiza la información mostrada en la pantalla de la interfaz.	
3. Cuando se vuelva a activar la carga de datos se mostrarán los datos antes ingresados.	
Postcondición	Se actualiza la información en pantalla.

Fuente: Elaboración propia.

4.2.7. Herramienta *Diary*

Por último, los casos de uso identificados para esta herramienta son tres. La tabla 4.7 muestra uno de ellos mientras que en la 4.4 se muestran más actividades de las que esta herramienta permite. Las actividades que no están implementadas son la selección un croquis y el ingreso de notas de texto. La tabla 4.16 muestra el único caso de uso adicional para el ingreso de registros.

Tabla 4.16. Caso de uso Editar registros.

Caso de uso: Editar registros	
Descripción	El propósito de este caso es editar los registros del diario.
Precondición	Ninguna.
Crear Registro	
1. El usuario activa el botón "Nuevo Registro".	
2. Se crea el registro, será identificado con el último número de la lista de registros.	
3. El número de registro se muestra en el cuadro de texto del panel.	
Postcondición	Se crea el registro.
Editar nota de texto	
1. El usuario modifica el texto asociado al registro.	
2. Se actualiza el texto y se almacena en el sistema.	
Postcondición	Se modifica el texto del registro.
Cargar imagen	
1. El usuario activa el botón "Añadir Imagen".	
2. El programa muestra la lista de imágenes, en formato JPG, en la carpeta "Imágenes" dentro del proyecto.	
3. El usuario selecciona la imagen que desea usar. No necesariamente debe estar dentro de la carpeta mostrada por defecto.	
4. Si el archivo no se encuentra dentro de la carpeta de imágenes, el programa crea una copia del archivo en esta carpeta.	
5. Se muestra la imagen cargada en el elemento Axes de la interfaz.	
Postcondición	Se muestra la imagen.

Eliminar Registro	
1. El usuario activa el botón "Eliminar Registro".	
2. El sistema borra el registro sin solicitar la confirmación del usuario.	
3. Los números asociados a los registros restantes son reasignados de manera que se mantenga la secuencia.	
Postcondición	Se elimina el registro seleccionado.

Fuente: Elaboración propia.

4.3. Requerimientos no funcionales

Estos requerimientos representan características generales y restricciones del proyecto. Dado que el programa que se desarrollará es sencillo en comparación a los softwares modernos solo se han identificado los siguientes.

Con respecto a la ejecución del programa, dado que una licencia original de MATLAB puede llegar a costar más de 2000 dólares, el programa debe ser capaz de funcionar sin necesidad de que el usuario cuente con una. Sin embargo, será necesario que el usuario cuente con un compilador del lenguaje de programación de MATLAB el cual es gratuito. El programa será compatible con versiones del compilador a partir de la versión 2015a.

En relación al tiempo de ejecución del programa, este debe ser mínimo y estará asociado a la capacidad de procesamiento del equipo en el cual se ejecute. Sobre la capacidad de almacenamiento de archivos y proyectos, estos estarán condicionados con la capacidad de almacenamiento del equipo.

La interfaz gráfica de usuario será implementada a través de la herramienta GUIDE que provee MATLAB. Por ello, las interfaces solo podrán contar con los elementos que dicha herramienta provee.

Finalmente, el sistema debe ser capaz de validar las acciones del usuario. Esto quiere decir que debe se debe informar, en la medida de lo posible, cuando alguna acción se esté realizando de manera incorrecta. Por ejemplo, cuando se ingrese texto en un campo donde debería ingresarse un número o tratar de cargar un archivo no válido.

CAPÍTULO 5: DISEÑO

En el presente capítulo se detalla el diseño del programa implementado. Un buen diseño facilita la posterior implementación del programa. Por ello, se podría decir que este es el capítulo más importante del proyecto. Se diseña una interfaz gráfica sencilla, considerando todos los elementos necesarios para garantizar que su uso sea intuitivo. La interfaz se complementa con funciones eficientes y variables estándar que son utilizadas en todo el programa. Es necesario detallar los estándares utilizados en el código del programa como nombres de variables, estructuras utilizadas, etc.

En primer lugar, se diseñan las interfaces gráficas de cada herramienta y la pantalla principal. Cada interfaz cuenta con distintos elementos como botones, cuadros de texto, sliders, etc. Al interactuar con estos elementos se produce un evento. Por ejemplo, pulsar un botón, ingresar un texto en un cuadro o mover un slider desencadenan eventos. Los eventos pueden ser asociados a funciones, de esta forma cuando el evento ocurre se ejecuta una serie de instrucciones. El código de las funciones debe seguir estándares definidos en base a las denominadas “buenas prácticas de programación” (Johnson, 2002).

Durante la ejecución del programa se crean variables que almacenan datos dentro del sistema y que existen mientras se ejecute el programa. Por otro lado, también se crean archivos que son almacenados dentro del disco duro del sistema y que existen aun cuando el programa deja de ejecutarse. La estructura de las variables y archivos también debe ser definida y respetada en todas las funciones.

A continuación, se muestra las interfaces de usuario diseñadas y los elementos que las conforman. Luego, se detalla las principales funciones implementadas, seguidas de la organización de los archivos dentro de la memoria de la computadora. Finalmente se presentarán los estándares definidos para el código del programa.

5.1. Interfaz gráfica de usuario (GUI)

La interfaz gráfica de usuario (GUI por sus siglas en inglés) es el medio a través del cual el usuario interactúa con el sistema. En términos más sencillos es la ventana que se muestra en pantalla cuando se ejecuta el programa. MATLAB permite crear interfaces gráficas a través de su herramienta GUIDE, la cual se puede apreciar en la Figura 5.1. En esta herramienta se muestra un campo en blanco sobre el cual se ubican elementos como botones y cuadros de texto.

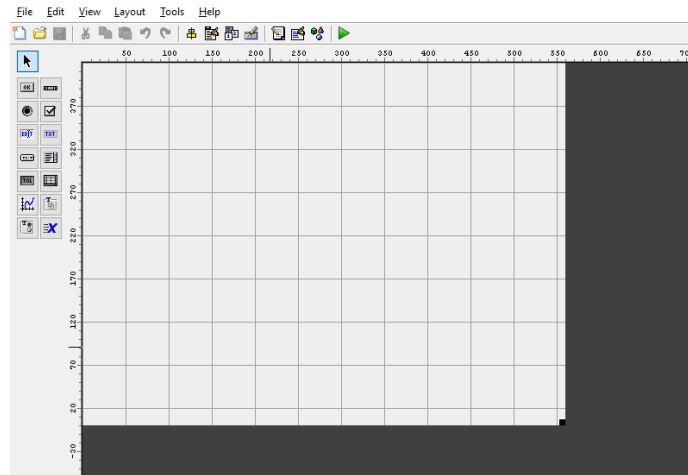


Figura 5.1. Herramienta GUIDE de MATLAB
Fuente: Herramienta GUIDE – MATLAB v2015a

Por un lado, existen elementos que desencadenan eventos cuando el usuario interactúa con ellos. Por ejemplo, un botón (Push Button), ver Figura 5.2, es un elemento cuyo evento es activado cuando el usuario lo pulsa al hacer clic con el mouse sobre él. La barra deslizador (Slider), que se muestra en la Figura 5.3, consiste de una barra que puede moverse dentro de un intervalo. El usuario puede mover la barra manteniendo pulsado el botón izquierdo del mouse sobre la barra y arrastrándola o usando los botones al extremo del slider. Un menú desplegable (Pop-up Menu) despliega múltiples opciones y permite seleccionar una de ellas. Para mostrar las opciones disponibles el usuario debe hacer clic sobre la flecha ubicada a la derecha del elemento, tal cual se muestra en la Figura 5.4. Finalmente, un botón radial (Radio Button), como el que se muestra en la Figura 5.5, permite marcar o desmarcar una casilla. Su evento se desencadena cuando se marca o desmarca la casilla al hacer clic sobre ella.

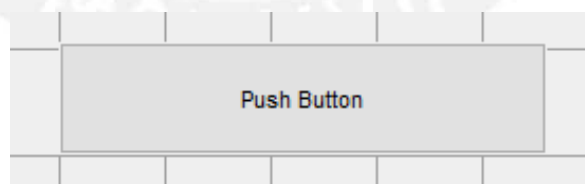


Figura 5.2. Elemento botón (Push Button)
Fuente: Herramienta GUIDE – MATLAB v2015a

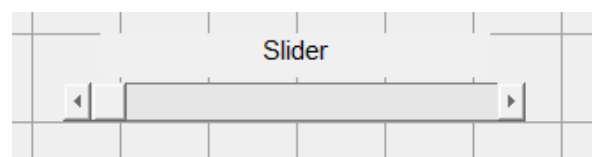


Figura 5.3. Elemento barra deslizador (Slider)
Fuente: Herramienta GUIDE – MATLAB v2015a

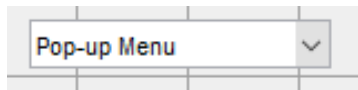


Figura 5.4. Elemento menú desplegable (Pop-up Menu)
Fuente: Herramienta GUIDE – MATLAB v2015a

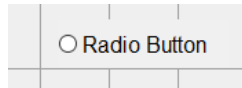


Figura 5.5. Elemento botón radial (Radio Button)
Fuente: Herramienta GUIDE – MATLAB v2015a

Por otro lado, se cuenta con dos elementos para mostrar texto en la pantalla. El texto estático (Static Text), que se muestra en la Figura 5.6 es un campo que contiene texto que no puede ser modificado directamente por el usuario. Este elemento no desencadena un evento y debe ser modificado por instrucciones de código. Por el contrario, el texto editable (Edit Text), ver Figura 5.7, si permite que el usuario edite el texto dentro del recuadro. En este caso, el evento se desencadena cuando el usuario modifica el texto editable.



Figura 5.6. Elemento texto estático (Static Text)
Fuente: Herramienta GUIDE – MATLAB v2015a

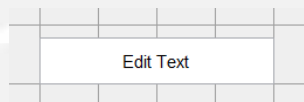


Figura 5.7. Elemento texto editable (Edit Text)
Fuente: Herramienta GUIDE – MATLAB v2015a

Otro elemento que no desencadena un evento es el Panel, el cual se muestra en la Figura 5.8. Este elemento permite agrupar otros elementos (incluso otros paneles) dentro de él. La principal función del panel es organizar los elementos dentro de la interfaz. De esta manera se puede modificar todos los elementos que pertenecen a un panel al mismo tiempo.



Figura 5.8. Elemento panel (Panel)
Fuente: Herramienta GUIDE – MATLAB v2015a

Por último, uno de los elementos más complejos es el elemento Ejes (Axes) que se muestra en la Figura 5.9. Este elemento consiste de un espacio sobre el cual se puede insertar imágenes o graficar funciones. Los elementos visualizados pueden ser de dos o tres dimensiones. Por defecto este elemento no desencadena ningún evento. Sin embargo; es posible definir eventos interactivos como por ejemplo cuando el usuario hace clic sobre un punto dentro del elemento.

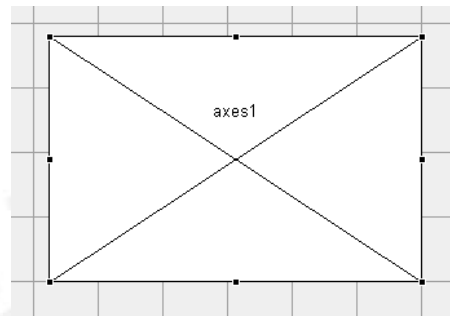


Figura 5.9. Elemento Ejes (Axes)
Fuente: Herramienta GUIDE – MATLAB v2015a

Además del evento que desencadenan, los elementos poseen propiedades. Las propiedades son campos que almacenan información asociada al elemento y que definen su comportamiento o apariencia. Algunas propiedades son comunes a todos los elementos, como por ejemplo la visibilidad, la posición, el estado de inactividad, entre otros. No obstante, otras propiedades dependen del tipo de elemento. Por ejemplo, el tipo y tamaño de letra son propiedades de un Texto Estático y no existen dentro de un Slider, el cual no posee un texto.

La propiedad más importante, y que es común en todos los elementos, se denomina "Tag". Esta propiedad es el nombre con el cual se identifica al elemento por lo que, el Tag de cada elemento es único.

Para el desarrollo del programa se diseñaron siete interfaces gráficas de usuario. Todas las interfaces fueron diseñadas desde la herramienta GUIDE de MATLAB y utilizando los elementos antes mencionados. A continuación, se detalla las interfaces diseñadas.

5.1.1. Pantalla Principal

La interfaz inicial del programa se divide en dos paneles, tal como se muestra en la Figura 5.10. En la mitad izquierda se encuentra el panel Menú Principal y en la mitad derecha el panel Menú de Proyecto.

En el primer panel se ubican cinco botones que ejecutan las funciones para Crear un proyecto, Guardar un proyecto, Abrir un proyecto, Exportar un proyecto e Importar un proyecto.

En el segundo panel se encuentran los campos de texto editable para ingresar el Lugar de Estudio y el Nombre de Usuario. El campo de Nombre de Proyecto no es editable, en este campo se muestra el nombre definido al momento de crear el proyecto. Dentro de este panel se encuentra el panel de herramientas. Cada herramienta cuenta con un Texto estático que identifica la herramienta, un menú desplegable en donde se elige el archivo que se desea abrir y un botón que abre el archivo elegido o crea uno nuevo si la opción del menú seleccionado es “Nuevo Archivo”.

Este panel también cuenta con un botón de nombre “Actualizar archivos”. Cuando el proyecto es abierto o se cierre la ventana de una de las herramientas, se vuelve a cargar la pantalla principal solo con dicho botón habilitado. El botón sirve para revisar las carpetas del proyecto y actualizar los archivos existentes de cada herramienta en sus respectivos menús desplegables.

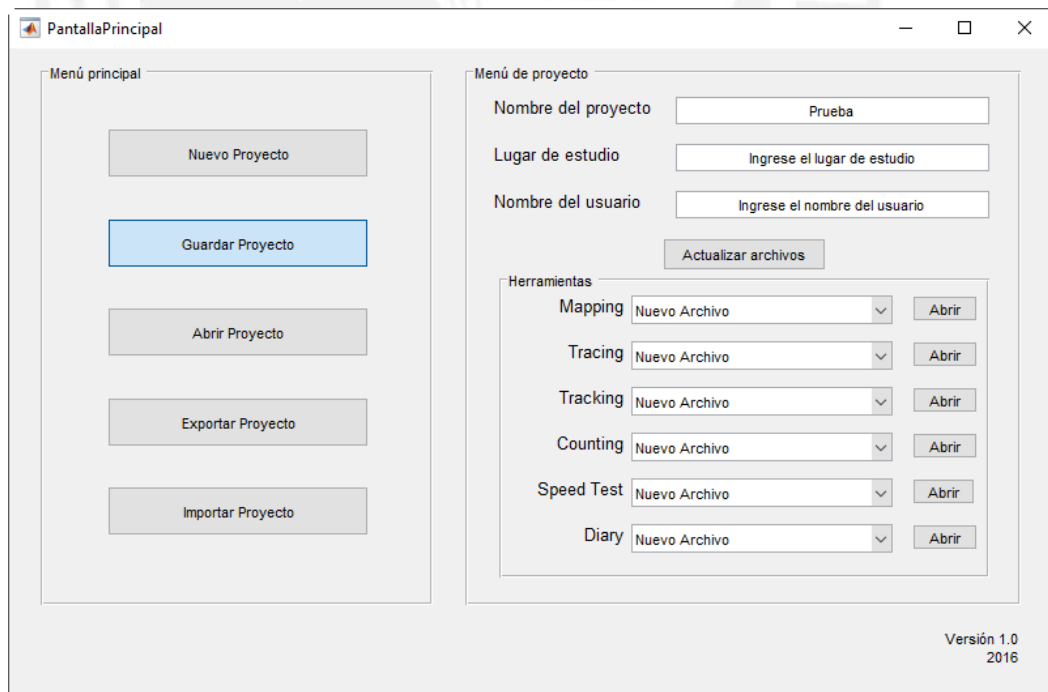


Figura 5.10. Interfaz Pantalla Principal
Fuente: Elaboración propia

Inicialmente solo se habilitan tres botones del panel Menú Principal. Cuando se activa un proyecto, ya sea al crear un nuevo proyecto o abrir un proyecto guardado, se habilitan los elementos restantes.

5.1.2. Herramienta *Tracing*

La interfaz de la herramienta *Tracing* se divide en cinco paneles, como se muestra en la Figura 5.11. En la parte superior se encuentra el panel Información del Proyecto. En la zona central, al extremo izquierdo se encuentra el panel Trazos. A su derecha se encuentra el panel Menú de Trazo. Bajo los dos últimos se encuentra el panel Menú. Finalmente, en la mitad derecha de la pantalla hay un panel sin un nombre asignado que contiene un elemento Axes que es sobre el cual se muestra el croquis de espacio analizado.

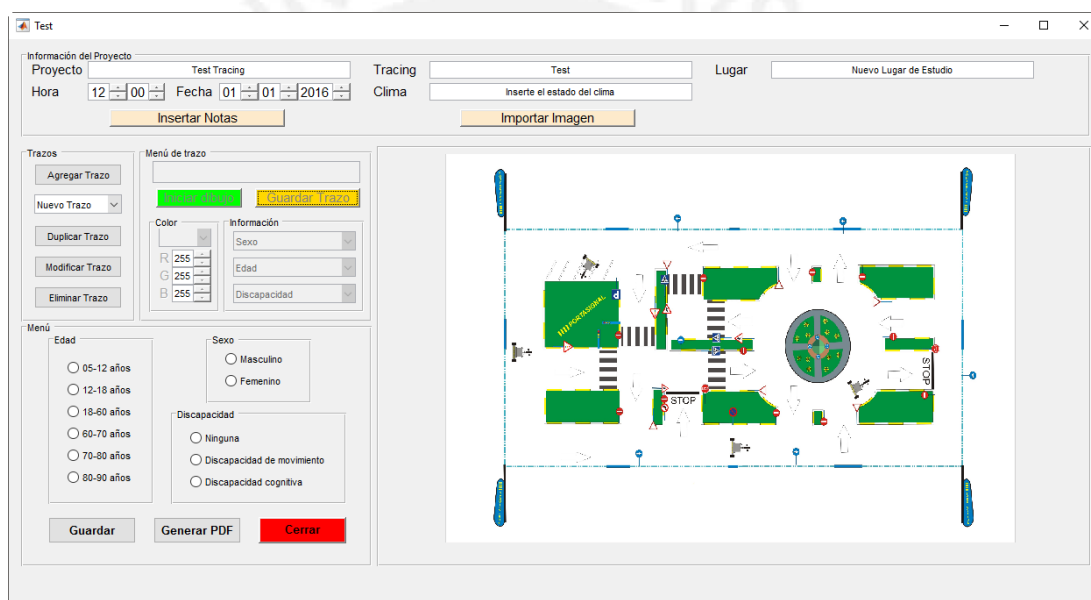


Figura 5.11. Interfaz *Tracing*
Fuente: Elaboración propia

En el panel Información del Proyecto se muestra el nombre del proyecto, el nombre del archivo, y el lugar de estudio ingresados en la Pantalla Principal. Además, este panel cuenta con un conjunto de sliders que permiten ingresar una hora y fecha. También existe un campo en donde el usuario puede definir las condiciones climáticas bajo las cuales se recolectó los datos. Finalmente, el panel cuenta con dos botones, el primero permite ingresar notas de texto que serán guardadas en el archivo y el segundo permite importar una imagen que luego será mostrada como croquis.

El panel Trazos cuenta con un menú desplegable en el cual se muestran los trazos existentes. También cuenta con cuatro botones que sirven para Crear, Modificar, Duplicar y Eliminar un trazo.

El panel Menú de Trazo se habilita cuando se ha activado un trazo. En la parte superior se muestra el nombre del trazo actualmente seleccionado. El panel cuenta con un subpanel llamado Color. En este panel se define el color del trazo a través de tres sliders que modifican las componentes rojo, verde y azul del color en formato RGB. También existe un subpanel llamado Información en donde se define los campos de Sexo, Edad y Discapacidad asociados al trazo. Por último, el panel cuenta con 2 botones: el botón Iniciar Dibujo habilita el programa para que el usuario pueda dibujar el trazo sobre el croquis y el segundo botón, de nombre Guardar Trazo, sirve para guardar el trazo actual y volver al Panel de Trazos. Luego de activar este último botón se deshabilita el panel Menú de Trazo

Tracing - T1

Nombre del Proyecto: P1
Nombre del Archivo: T1

Ingrese sus notas

Lugar de estudio: Nuevo Lugar de Estudio
Nombre del Usuario: Nuevo Usuario

Condiciones climáticas:
Fecha: 01/01/2016
Hora: 12:00

Filtros aplicados:
Edad:
Sexo:
Discapacidad:

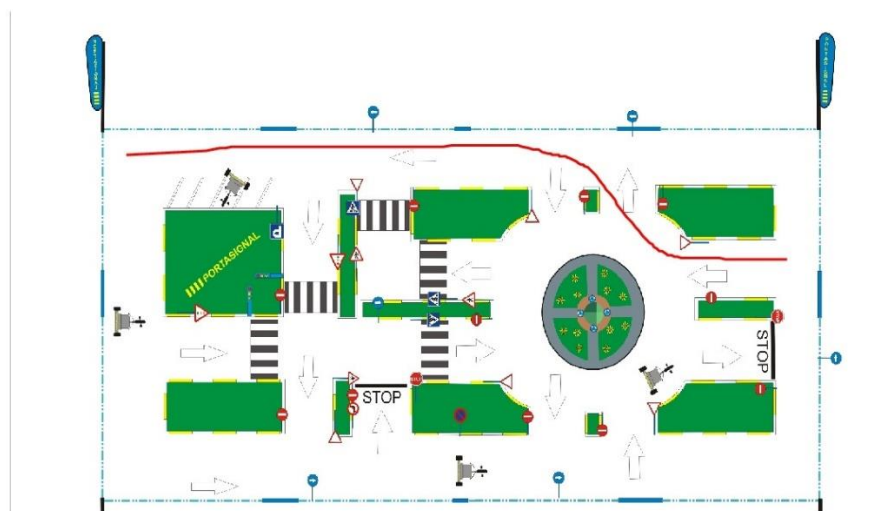


Figura 5.12. PDF generado por la herramienta Tracing.
Fuente: Elaboración propia

Al lado derecho de la pantalla se encuentra el panel del croquis. Este panel cuenta con un único elemento: un Axes. Sobre el plano se grafican los trazos ingresados por el usuario.

Finalmente se tiene el panel Menú. Dentro de éste se ubican tres subpaneles en los cuales se define la información de sexo, edad y discapacidad de los trazos que se muestran sobre el croquis. Para ello, dentro de cada subpanel se encuentra una serie Radio Button asociados a cada estado. Bajo los subpaneles mencionados se encuentran tres botones. El primer botón permite Guardar el archivo para poder ser abierto desde la Pantalla Principal. El segundo botón genera un archivo PDF con la información mostrada en la figura en ese momento, como el que se muestra en la Figura 5.12. El último botón cierra la ventana de la interfaz y vuelve a abrir la Pantalla Principal que fue cerrada antes de abrir la interfaz de Tracing.

5.1.3. Herramienta Mapping

Esta interfaz cuenta con cuatro paneles tal como se muestra en la Figura 5.13. En la parte superior se encuentra el panel de información del proyecto el cual es estándar para todas las herramientas.

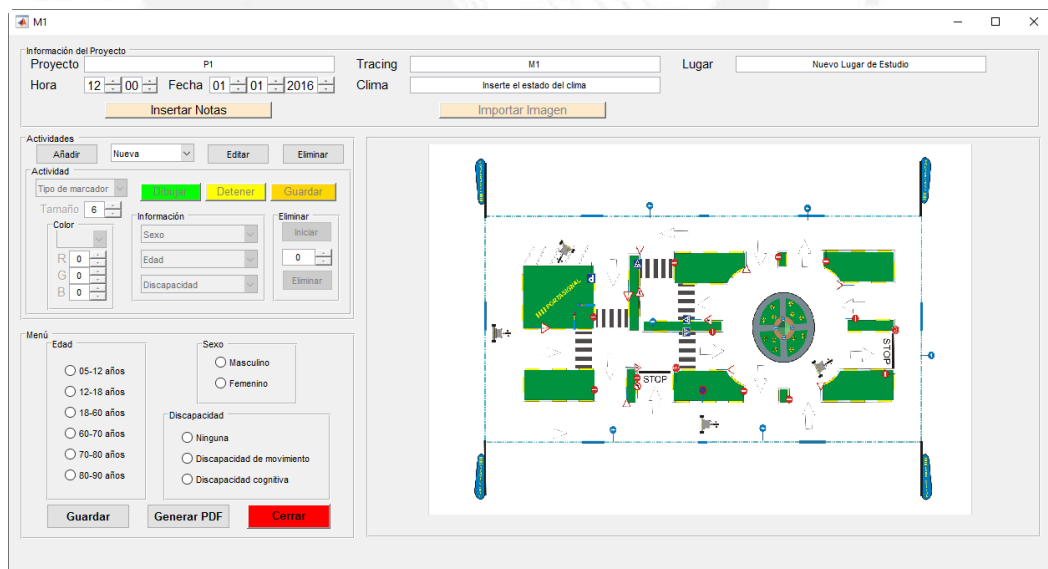


Figura 5.13. Interfaz de la herramienta Mapping.
Fuente: Elaboración propia

En la mitad derecha se encuentra el elemento Axes sobre el cual se ha dibujado la imagen seleccionada por el usuario. En este panel se irá mostrando los puntos que el usuario ingrese. Además, se mostrará la

leyenda con las actividades definidas. Dicha leyenda puede ser reubicada usando el mouse.

En la mitad izquierda se ubican los paneles de actividades y el menú de visualización. Dentro del panel de actividades se ubican tres botones, un menú desplegable y un subpanel. Los botones sirven para crear una nueva actividad, editar una actividad existente o eliminar la actividad que se encuentre seleccionada en el menú desplegable. Cuando se crea o edita una actividad se habilita el subpanel Actividad.

En el panel Actividad, se cuenta con un menú desplegable para seleccionar el tipo de marcador para los puntos que pertenezcan a la actividad habilitada. También se encuentra un slider que permite modificar el tamaño del marcador y finalmente un subpanel, similar a la herramienta anterior, para seleccionar el color usado. Luego, se cuenta con menús desplegables para seleccionar características de sexo, edad y presencia de discapacidades. Para poder iniciar a dibujar los puntos se utiliza el botón “Dibujar” y para dejar de dibujarlos se usa el botón “Detener”. El panel “Eliminar” permite eliminar puntos de la actividad actual a través de un slider que itera sobre los puntos existentes y un botón para eliminar el punto seleccionado. Por último, el botón “Guardar” salva la información añadida y vuelve a habilitar el panel Actividades.

El último panel donde se ubica tres bloques de casillas de marcación sirve para filtrar las actividades que se muestran en el eje. De esta manera el usuario puede seleccionar solo aquellas que sean de su interés. En el mismo panel se ubican los botones para guardar el archivo, imprimir un reporte en formato PDF y cerrar la ventana. Cuando se cierra la ventana se abre la pantalla principal.

Mapping - M1

Nombre del Proyecto: P1
Nombre del Archivo: M1

Lugar de estudio: Nuevo Lugar de Estudio
Nombre del Usuario: Nuevo Usuario

Condiciones climáticas:
Fecha: 02/06/2019
Hora: 12:00

Filtros aplicados:
Edad:
Sexo:
Discapacidad:

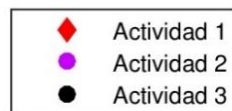
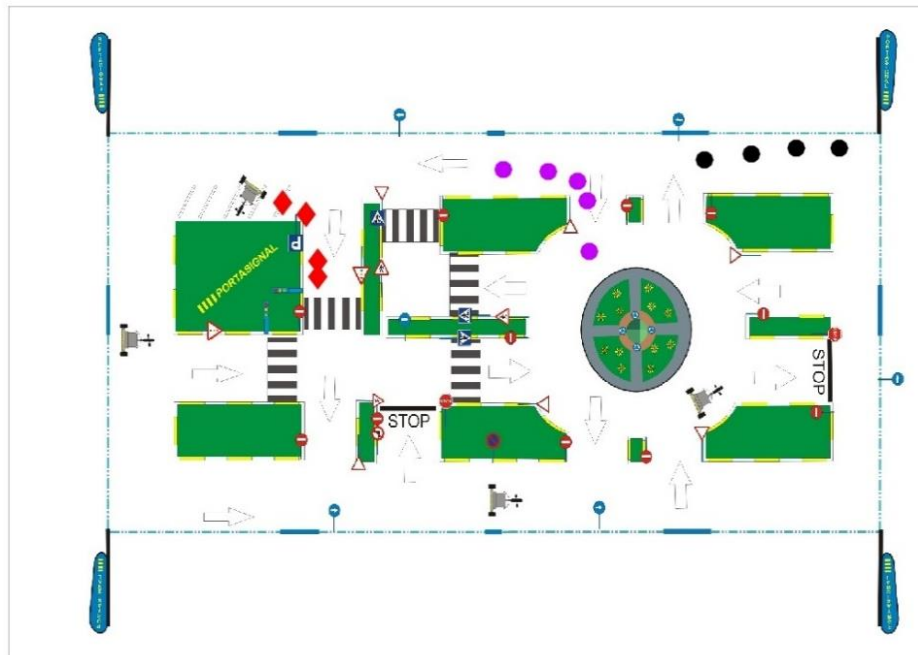


Figura 5.14. PDF generado por la herramienta Mapping.
Fuente: Elaboración propia

5.1.4. Herramienta *Tracking*

Esta interfaz cuenta con cinco paneles. En la parte superior se encuentra el panel de información general del proyecto. En la mitad derecha se ubica el elemento Axes sobre el que se muestra el croquis importado. Sobre esta imagen se dibujará la trayectoria y los puntos del registro. La figura 5.15 muestra la interfaz elaborada.

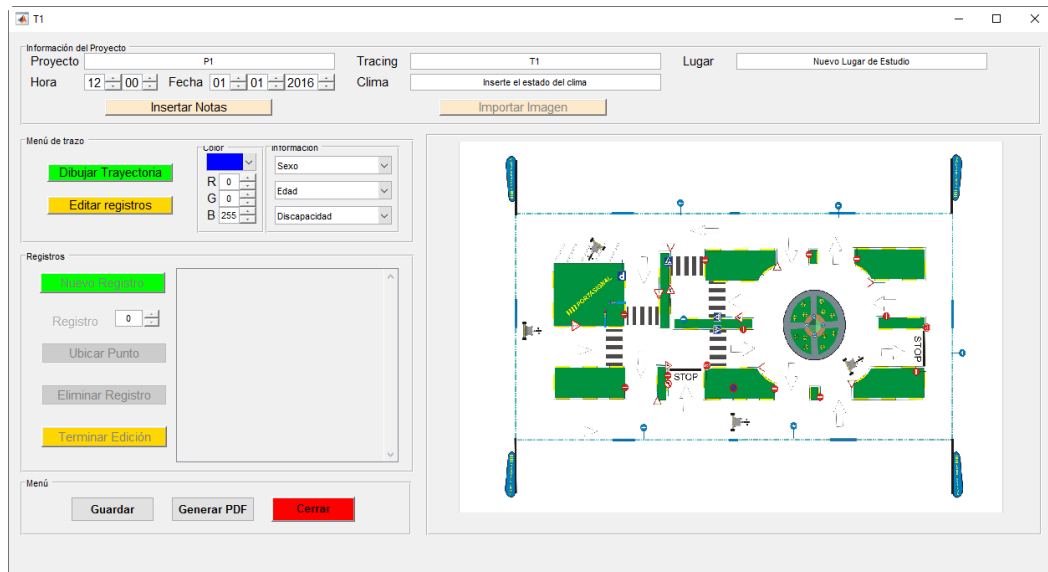


Figura 5.15. Interfaz de la herramienta Tracking.
Fuente: Elaboración propia

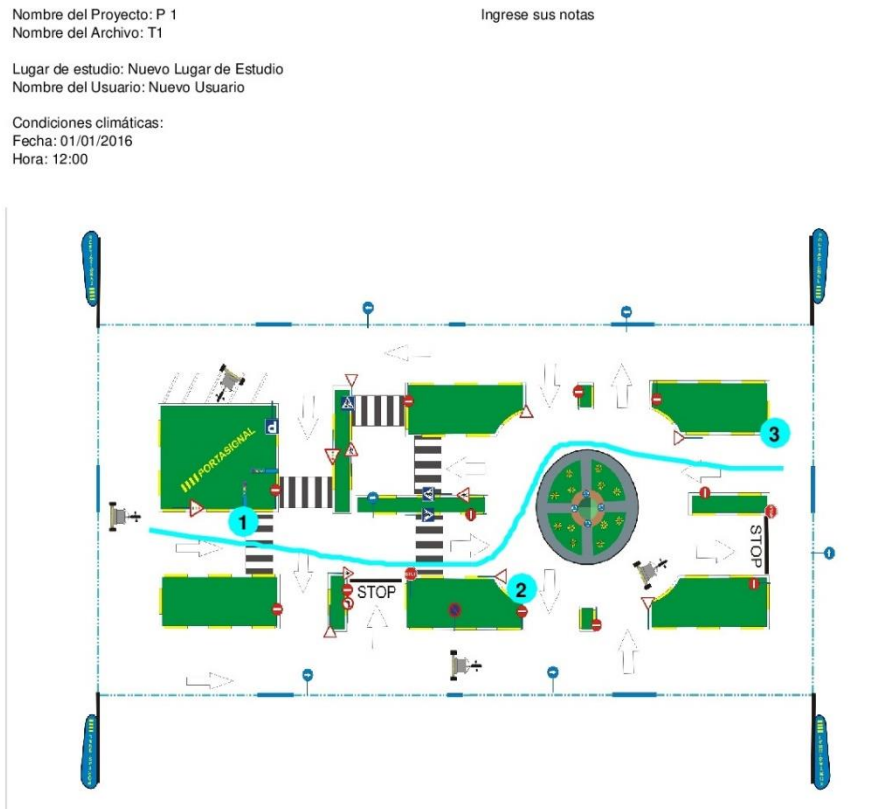
En la mitad izquierda se ubican tres paneles. Primero, el panel Menú de Trazo contiene un botón para dibujar un trazo sobre el croquis que representa la trayectoria analizada. En el mismo panel se encuentra un subpanel para cambiar el color de la trayectoria y un subpanel para definir las características de sexo, edad y presencia de discapacidades. Finalmente, con el botón Editar Registros se habilita el panel Registros y se deshabilita el panel actual.

En el panel de registros se encuentra cuatro botones, un slider y un cuadro de texto editable. El primer botón sirve para crear un nuevo registro. El slider permite desplazarse entre los registros existentes. El segundo botón sirve para ubicar un punto en el croquis que está asociado al registro. El punto estará etiquetado con el número del registro. El tercer botón elimina el registro actual y el cuarto botón finaliza la edición y deshabilita el panel Registros. Al finalizar, en el croquis se muestran todos los puntos existentes además de la trayectoria.

Por último, en el panel Menú solo se encuentran tres botones que sirven para guardar el archivo, generar el reporte en formato PDF y cerrar la ventana. Esta herramienta crea dos archivos PDF: en el primero que tiene la extensión “croquis” se presenta la información general del proyecto y el croquis con la trayectoria y los puntos de los registros; en el segundo, que tiene extensión “registros” se muestra el texto de todos los registros en orden creciente. En la Figura 5.16 se muestra un ejemplo de

un reporte generado. Similar a las herramientas anteriores, al cerrar la ventana se abre la pantalla principal del programa.

Tracking - T1



Tracking - T1

Registro 1: Se detuvo a revisar su teléfono

Registro 2: Se encontró con un conocido y conversaron por varios minutos.

Registro 3: Espero hasta poder cruzar la calle

Figura 5.16. PDF generados por la herramienta Tracking.
Fuente: Elaboración propia

5.1.5. Herramienta *Counting*

Esta interfaz cuenta con tres paneles. En la parte superior se encuentra el panel de información general del proyecto. En la mitad izquierda se encuentra el elemento Axes sobre el cual se dibuja el croquis cargado y

en la mitad derecha el panel de visualización en donde se muestra la interpretación de los valores ingresados. La interfaz se muestra en la Figura 5.17.

Dentro del primer panel se ubican, además del Axes, dos sliders para definir la hora de inicio de análisis y también dos cuadros de texto editables para ingresar la duración y cantidad de los periodos. También se cuenta con dos botones para dibujar la sección en la que se tomaron los datos y para abrir una ventana para ingresar los valores numéricos.

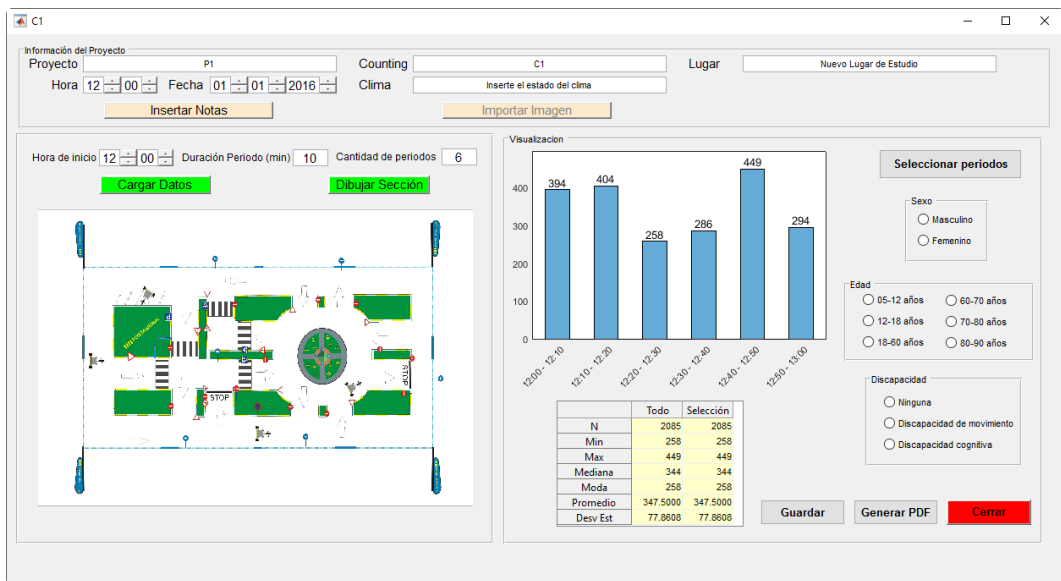


Figura 5.17. Interfaz de la herramienta Counting.
Fuente: Elaboración propia

Cuando se activa el botón “Cargar Datos” se abre la ventana del mismo nombre que muestra en la Figura 5.18. En ella, se cuenta con una tabla para ingresar los flujos registrados manualmente. La tabla cuenta con cinco columnas correspondientes a la cantidad de personas, el periodo en el que fueron registrados y tres campos para definir las características de las personas. Además, existen tres botones: el primero para cargar una hoja de Excel, el segundo para añadir una fila en la tabla y el tercero para finalizar la carga de datos y regresar a la interfaz de la herramienta.

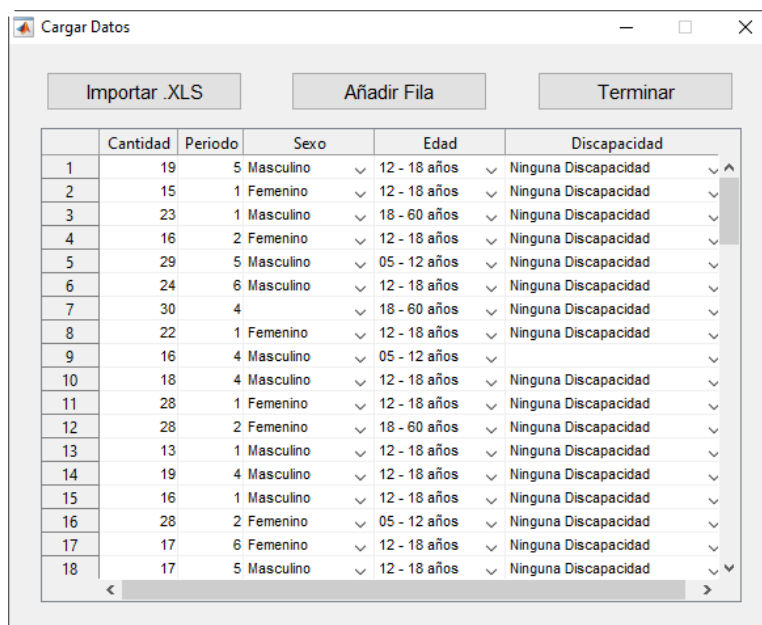


Figura 5.18. Ventana para ingresar datos – Herramienta Counting.
Fuente: Elaboración propia

En la mitad derecha se encuentra el panel de visualización. Cuando se ha ingresado los datos de los flujos registrados en cada periodo se actualiza el histograma presente los datos estadísticos mostrados en la tabla del panel. Similar a las herramientas Tracing y Mapping, se cuenta con tres conjuntos de casillas de marcación para filtrar los datos mostrados. Además, el botón “Seleccionar Periodos” añade un criterio más al seleccionar los periodos de interés.

En el mismo panel se ubican los botones para guardar el archivo, generar el reporte PDF y cerrar la ventana. En este caso, el reporte, Figura 5.19, incluye el croquis con la sección dibujada, la tabla de parámetros estadísticos y el histograma según los criterios seleccionados.

Counting - C1

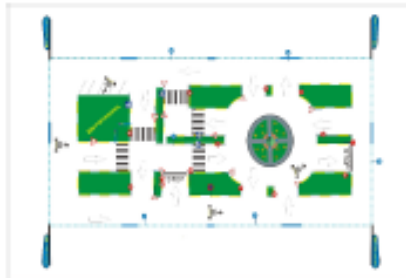
Nombre del Proyecto: P1
Nombre del Archivo: C1

Ingrese sus notas

Lugar de estudio: Nuevo Lugar de Estudio
Nombre del Usuario: Nuevo Usuario

Condiciones climáticas:
Fecha: 01/01/2016
Hora: 12:00

Filtros aplicados:
Edad:
Sexo:
Discapacidad:



	Todo	Selección
N	2085	2085
Min	258	258
Max	449	449
Mediana	344	344
Moda	258	258
Promedio	347.5000	347.5000
Dev Est	77.8608	77.8608

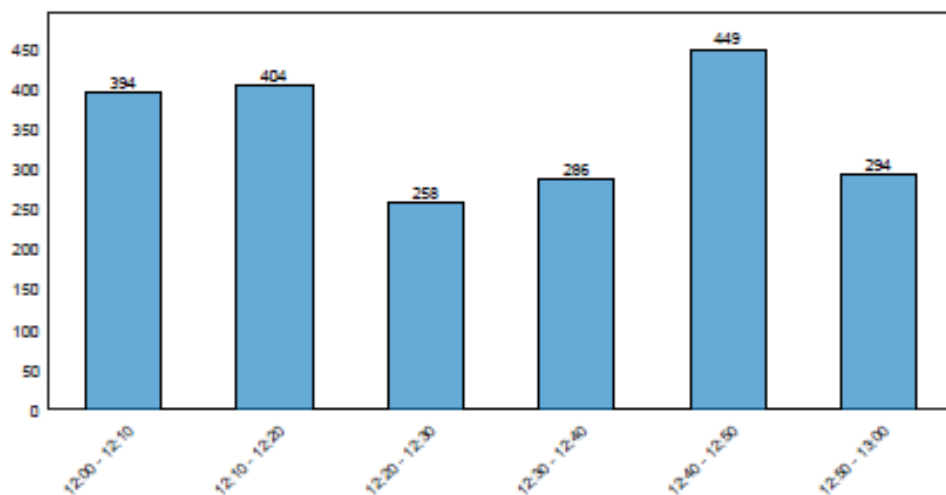


Figura 5.19. PDF generados por la herramienta Counting.
Fuente: Elaboración propia

5.1.6. Herramienta *Speed Test*

Esta interfaz es similar a la herramienta anterior. La distribución de los paneles es la misma, Figura 5.20. Por ello, solo se detallará los elementos que difieren con la ventana de la herramienta Counting.

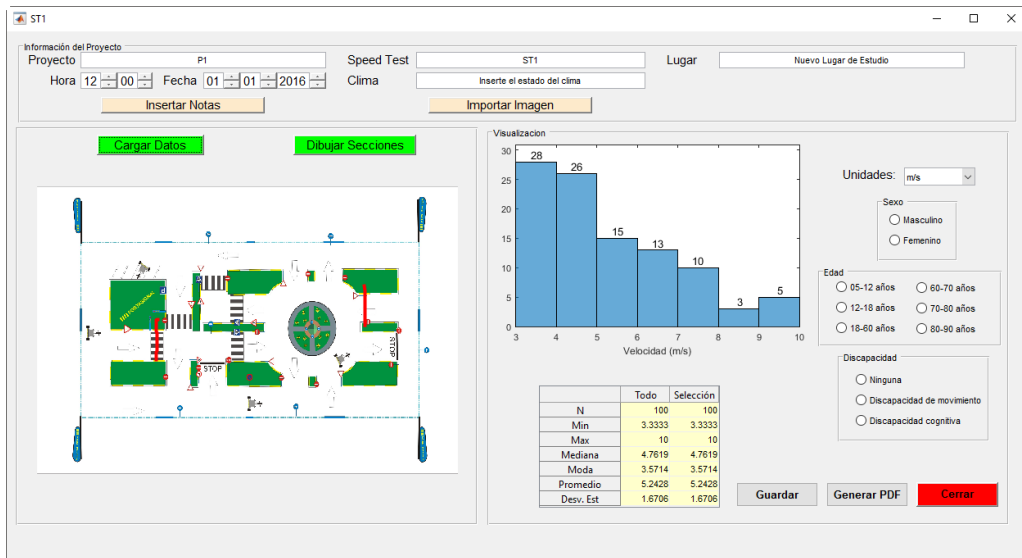


Figura 5.20. Interfaz de la herramienta Speed Test.
Fuente: Elaboración propia

En primer lugar, en el panel de Axes solo existe dos botones para cargar los datos y dibujar las secciones. Como esta herramienta analiza las velocidades empleadas para desplazarse de un punto a otro, se debe dibujar dos secciones sobre el plano. En la pantalla “Cargar Datos”, Figura 5.21, se añade un campo adicional para ingresar la distancia, en metros, que separa las secciones dibujadas.

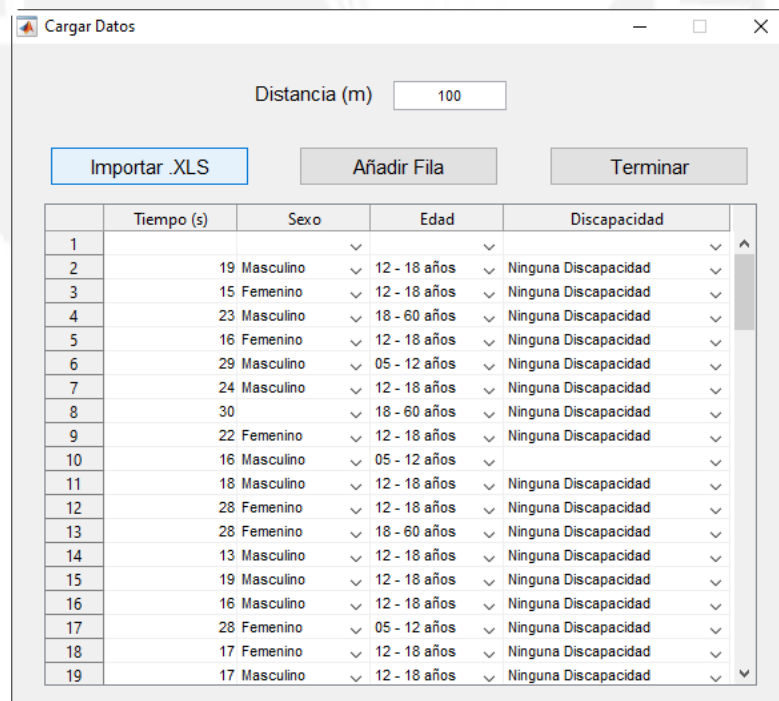


Figura 5.21. Ventana para ingresar datos – Herramienta Speed Test.
Fuente: Elaboración propia

Finalmente, en el panel Visualización ya no está el botón para seleccionar los periodos de interés, en su lugar se cuenta con un menú desplegable para seleccionar las unidades en las que se muestra los datos: metros por segundo o kilómetros por hora. El histograma ya no dividido según el periodo de análisis, sino según la frecuencia de las velocidades presentes. El PDF generado también tiene la misma estructura que el generado por la herramienta Counting, Figura 5.22.

Speed Test - ST1

Nombre del Proyecto: P 1
 Nombre del Archivo: ST1

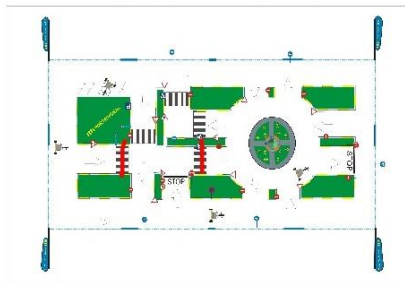
Ingrese sus notas

Lugar de estudio: Nuevo Lugar de Estudio
 Nombre del Usuario: Nuevo Usuario

Condiciones climáticas:
 Fecha: 01/01/2016
 Hora: 12:00

Filtros aplicados:
 Edad:
 Sexo:
 Discapacidad:

Distancia = 100 m



	Todo	Selección
N	100	100
Min	3.3333	3.3333
Max	10	10
Mediana	4.7619	4.7619
Moda	3.5714	3.5714
Promedio	5.2428	5.2428
Desv. Est	1.6706	1.6706

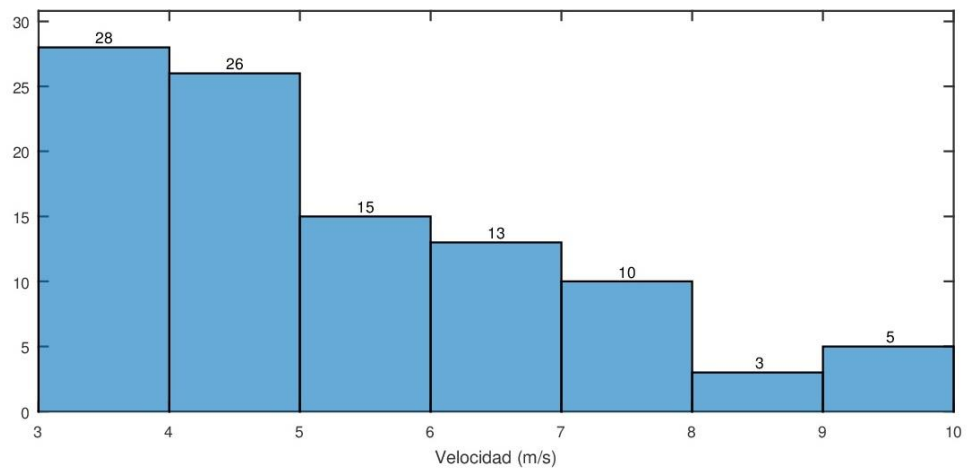


Figura 5.22. PDF generados por la herramienta Counting.
 Fuente: Elaboración propia

5.1.7. Herramienta *Diary*

La interfaz de esta herramienta cuenta con cuatro paneles, Figura 5.23. En la parte superior se encuentra el panel de Información del Proyecto; sin embargo, a diferencia de las demás herramientas, en este panel ya no están los botones para agregar notas de texto ni importar una imagen.

En la parte izquierda de la pantalla se encuentra el panel de Registros. En este panel se cuenta con un botón para crear un nuevo registro, un slider para desplazarse entre los registros existentes, un botón para añadir una imagen al registro y un botón para eliminarlo. A cada registro se le puede asignar una fecha y hora. Finalmente, se tiene un cuadro de texto editable en donde el usuario puede redactar las notas que crea conveniente. En la parte derecha de la pantalla se encuentra un elemento Axes que muestra la imagen asociada al registro activo.

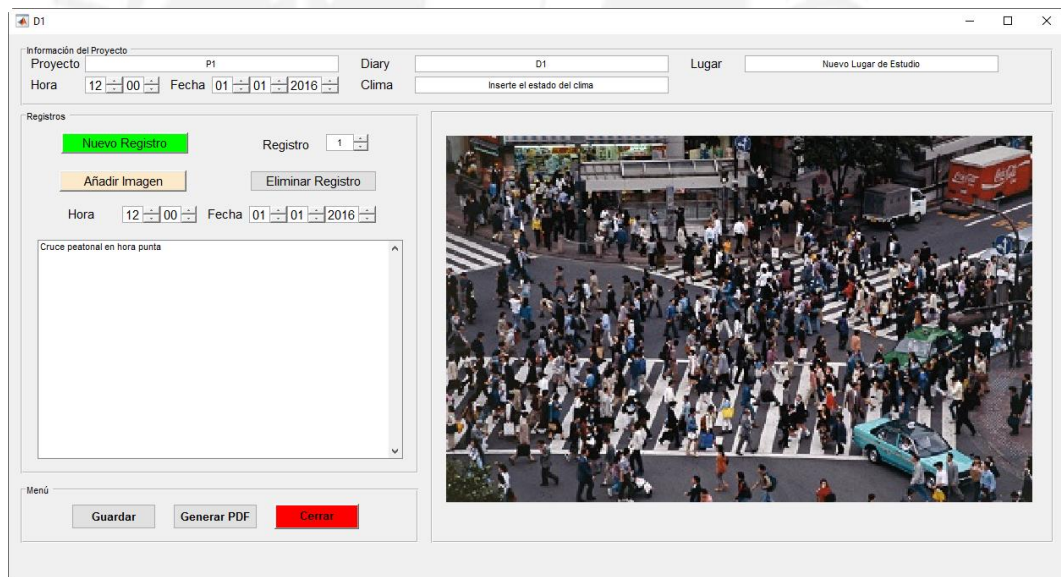


Figura 5.23. Interfaz de la herramienta Diary.
Fuente: Elaboración propia

Finalmente, igual que en todas las herramientas se cuenta con tres botones para guardar, cerrar y generar un reporte en formato PDF, Figura 5.24. En este caso, el reporte incluye tres imágenes por hoja y el programa crea un archivo por cada hoja que sea necesaria para cubrir todos los registros.

Nombre del Proyecto: P1
Nombre del Archivo: D1

Condiciones climáticas:
Fecha: 01/01/2016
Hora: 12:00

Lugar de estudio: Nuevo Lugar de Estudio
Nombre del Usuario: Nuevo Usuario

01/01/2016 12:00

Registro: 1
Fecha: 01/01/2016
Hora: 12:00

Cruce peatonal en hora punta.



Registro: 2
Fecha: 01/01/2016
Hora: 14:00

Zona de tránsito peatonal



Registro: 3
Fecha: 01/01/2016
Hora: 12:00

Ruta de tránsito peatonal exclusivo



Figura 5.24. PDF generados por la herramienta Diary.
Fuente: Elaboración propia
Fotografías obtenidas de Gehl y Svarre (2013)

Luego de presentar todas las interfaces diseñadas es importante realizar algunas acotaciones generales. La interfaz principal funciona como gestora de las demás interfaces ya que es la única forma de ejecutar las demás herramientas. Como se habrá podido notar, existen varios elementos en común entre las herramientas como el panel de información de proyecto o los paneles para la visualización. Estas similitudes facilitan la codificación del programa ya que se puede utilizar las mismas funciones con pequeñas modificaciones. Parte de un buen diseño consiste en tener funciones que sean reutilizables.

5.2. Funciones utilizadas

La lista de funciones existentes en todo el programa es bastante extensa. La mayoría de las funciones son sencillas y sirven para cambiar las propiedades de los elementos de la interfaz. Por ejemplo, actualizar cuadros de texto, habilitar y deshabilitar elementos, entre otros. Algunas de las funciones utilizadas no fueron diseñadas como parte del proyecto. Estas funciones, creadas por personas externas, se encuentran publicadas en el repositorio de MATLAB (2016) o foros en línea donde se comparte código de manera gratuita. Todas las funciones obtenidas a través de este medio serán presentadas más adelante en esta sección.

Se puede dividir las funciones en tres categorías según su objetivo: para el ingreso de datos, para la manipulación de los datos y para la visualización de los resultados. Siguiendo esta clasificación, a continuación, se presentan las funciones más importantes en el programa.

5.2.1. Ingreso de datos

Este tipo de funciones sirven para solicitar información al usuario directa o indirectamente. El ingreso de datos puede ser a través del teclado, utilizando el mouse o a través de archivos externos que pueden ser imágenes, hojas de cálculo, entre otros. A continuación, se presentan las funciones utilizadas.

a. uipickfiles.m

Esta es aplicación, que cuenta con una interfaz propia que se muestra en la Figura 5.25. La aplicación fue diseñada por Douglas Schwarz (2006) y permite obtener la ruta de múltiples archivos y carpetas en el equipo. Esta aplicación también permite definir la cantidad de archivos que el usuario debe seleccionar y el formato de los mismos. Esta aplicación resultó muy útil en el programa ya que permitía seleccionar las imágenes para importar, como croquis o en el diario; elegir los proyectos para importar o abrir; y finalmente seleccionar los archivos de Excel para usar en las herramientas Counting y Speed Test.

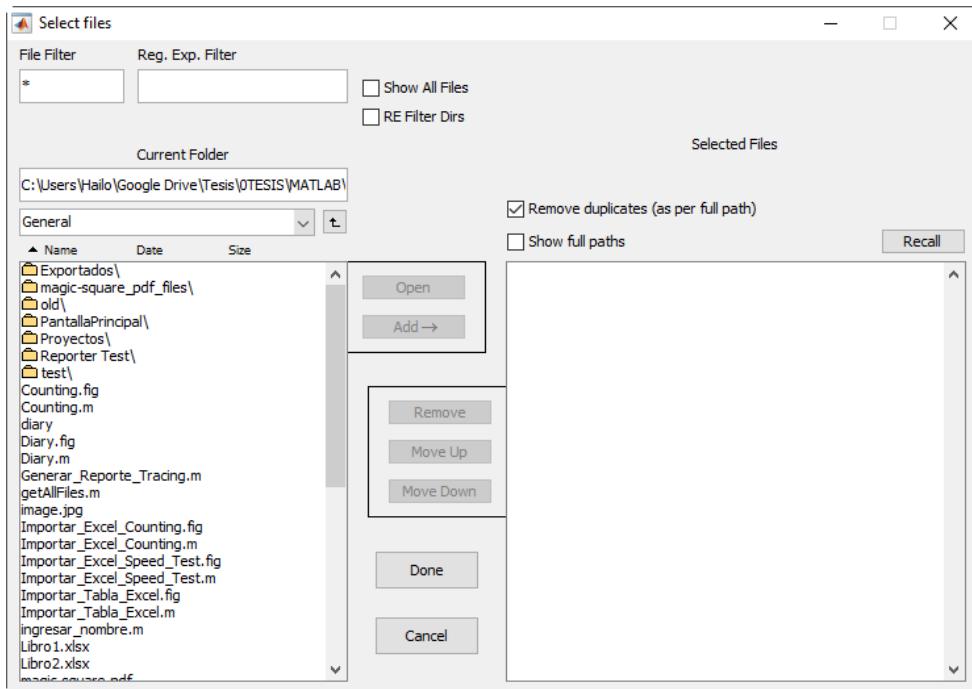


Figura 5.25. Ventana herramienta uipickfiles.
Fuente: Schwarz (2006)

b. `ingresar_nombre.m`

Esta función fue diseñada para este proyecto y su objetivo es solicitar una cadena de caracteres al usuario a través de la función nativa de MATLAB `inputdlg`. La función valida que el texto ingresado contenga solo números y letras y también eliminar los espacios en blanco excedentes al inicio, final o entre las palabras.

Esta función es utilizada en múltiples ocasiones durante el programa cada vez que se solicita ingresar un nombre. Ya sea para un nuevo proyecto, para crear archivos de las herramientas, nombres para trazos (Tracing) o actividades (Mapping).

c. `startDragFcn`, `draggingFcn`, `stopDragFcn`

Las tres funciones son *callbacks*, es decir que son eventos que se ejecutan cuando el usuario interactúa con un elemento de la interfaz, en este caso un Axes. Los eventos corresponden a hacer clic sobre la pantalla, mover el cursor manteniendo el botón izquierdo presionado y soltar el botón respectivamente. Con estas funciones se puede obtener la posición del cursor a medida que se mueve y con esos datos graficar trazos o puntos. Estas funciones son utilizadas en todas las interfaces excepto la herramienta Diary.

La función utilizada en el programa fue desarrollada con base en la función publicada en el foro Stack Overflow por el usuario Sawsan (2012).

d. xlsread

Esta función, nativa de MATLAB, es utilizada para leer datos desde un archivo de Excel. La función obtiene todos los datos en la primera hoja del archivo, por defecto, o de la hoja que el usuario especifique. Lo útil de esta función es que no necesita abrir el archivo para poder leer su contenido. Esta función se utiliza en las herramientas Counting y Speed Test para cargar datos desde una hoja de cálculo.

5.2.2. Manipulación de datos

Esta categoría contiene las funciones que son utilizadas para manipular los datos o archivos mientras el programa se ejecuta. Como se explicó anteriormente, el objetivo del programa no es realizar una manipulación compleja de los datos ingresados sino, mostrarlos y administrarlos de una manera eficiente y ordenada.

Los únicos cálculos realizados son en las herramientas Counting y Speed Test en donde se genera un histograma y se presentan parámetros estadísticos sencillos. Todos los cálculos son realizados por funciones nativas de MATLAB. Con respecto a la administración de los archivos se utilizaron las funciones que se presentan a continuación:

a. savefig, openfig

Ambas son funciones nativas de MATLAB y sirven para guardar y abrir archivos, respectivamente, en formato FIG. Este formato es utilizado por MATLAB para almacenar elementos gráficos como la interfaz de una aplicación. El programa utiliza las funciones para guardar y abrir los archivos de los proyectos de manera que se mantengan las modificaciones y datos ingresados por el usuario.

b. getAllFiles.m

Esta función sirve para obtener la lista de archivos que existen dentro de una carpeta especificada como parámetro de entrada. Esta

función es utilizada para actualizar la lista de archivos de cada herramienta en la Pantalla Principal con el objetivo de no omitir ninguno. Otra aplicación para esta función es evitar sobrescribir archivos existentes cuando se use el mismo nombre.

La función utilizada en el programa es una modificación de la función publicada en el foro Stack Overflow por el usuario Oz Radiano (2014).

5.2.3. Visualización de datos

Esta última categoría de funciones se encarga de generar las salidas que produce el programa según los datos ingresados. Una de las salidas son los archivos del programa que son guardados con la función `savefig` detallada en la sección anterior. La otra salida son los reportes generados por todas las herramientas. A continuación, se resumen las funciones utilizadas.

a. `figure`, `uicontrol`, `uitable`

Estas funciones son nativas de MATLAB y permiten generar elementos gráficos. La función `figure` crea una figura que, en términos sencillos, es un lienzo en blanco sobre el cual se pueden añadir los elementos mencionados en la sección 5.1. La función `uicontrol` se encarga de crear dichos elementos. Esta función recibe como parámetro el tipo de objeto que va a crear que puede ser un texto estático, un Axes, entre otros. Por otro lado, la función `uitable` crea una tabla en la figura con los datos que son especificados como parámetros de entrada.

Estas funciones son utilizadas para crear los reportes PDF. Primero se crea una figura en blanco. Luego se añade los elementos como cuadros de texto, Axes y tablas. Finalmente se modifica el contenido de dichos elementos con la información que se desea mostrar.

b. `print`

Esta función convierte una figura en un archivo PDF y es nativa de MATLAB. El archivo creado incluye todos los elementos que la figura contenga. Utilizando esta función se crean los reportes según las

figuras creadas. Esta función permite especificar la ruta en la que el archivo será creado. Además del formato PDF, esta función también puede crear imágenes en formato JPG o TIFF; sin embargo, estas funcionalidades no son utilizadas en este proyecto. Esta función es utilizada en todas las herramientas.

5.3. Organización de los archivos

El programa es almacenado dentro de una sola carpeta, que para efectos de la explicación denominaremos “General”. El usuario puede ubicar dicha carpeta en cualquier lugar dentro de su equipo. Dentro de la carpeta se encuentra los archivos del programa que incluyen los códigos y figuras utilizadas. Estos archivos no deben ser modificados ni reubicados bajo ningún motivo.

Dentro de la carpeta se encuentran dos subcarpetas de nombres “Exportados” y “Proyectos”. En la primera carpeta se guardarán los proyectos que sean exportados desde la pantalla principal del programa. En la segunda carpeta se almacenarán los archivos correspondientes a todos los proyectos creados.

Dentro de la carpeta “Proyectos” se contará con una carpeta para cada proyecto existente. El nombre de la carpeta coincidirá con el nombre del proyecto. Dentro de dicha carpeta se guardará un único archivo de nombre “PantallaPrincipal.fig” el cual es utilizado para abrir el proyecto. Además del archivo mencionado, dentro de la carpeta del proyecto se creará siete carpetas: seis son asociadas a cada una de las herramientas implementadas y la última, de nombre “Imágenes” sirve para almacenar las imágenes que sean cargadas por el programa. Dentro de cada carpeta de las herramientas se contará con múltiples archivos, de acuerdo a los registros creados. Cada archivo sirve para poder abrirlo y editarlo desde el programa. Finalmente, además de los archivos se creará una carpeta de nombre “PDF” en donde se almacene los reportes generados por el usuario a través del programa.

5.4. Estándares de código

El programa está destinado para un público que no necesariamente cuente con experiencia en programación o MATLAB. Tampoco será necesario que el usuario acceda, modifique ni entienda el código implementado. Sin embargo, esto no significa que la calidad del código sea dejada de lado. Por ello, al momento de la implementación es necesario seguir ciertas convenciones que

permitan que el código sea entendible para cualquier persona con conocimiento del lenguaje. Otra ventaja de tener un código “limpio” es que facilita su futuro mantenimiento o ampliación. Existen múltiples convenciones que son aplicables a cualquier lenguaje de programación, así como existen algunas que solo funcionan para el lenguaje de MATLAB. A continuación, se presentan las consideraciones tomadas en cuenta al momento de redactar el código según la recopilación realizada por Johnson (2002).

Todas las variables utilizadas son identificadas por un nombre. Es importante que el nombre asignado refleje el significado del dato que almacena la variable. Por ello, se recomienda evitar nombres genéricos como x o n y utilizar nombres que brinden información por sí solos como *ruta_proyecto* o *trazo_actual*. Tampoco deben utilizarse nombres de variables muy extensos ya que sobrecargan el código y dificultan la redacción. Una convención aceptada en cualquier lenguaje de programación es utilizar las letras minúsculas como i , j o k para identificar una variable iterada. Estas consideraciones no solo aplican para el nombre de variables, sino también para identificar constantes y funciones. Una recomendación mencionada por Johnson (2002) es redactar todos los nombres de las variables en inglés ya que es el lenguaje aceptado a nivel internacional para el desarrollo de software; sin embargo, en el proyecto se ha preferido utilizar nombres en español para facilitar su entendimiento.

La herramienta más útil para facilitar el entendimiento del código son los comentarios. Los comentarios son líneas de texto en el código que no son ejecutadas por el programa. Cada lenguaje de programación tiene una manera particular de identificar un comentario. En el caso de MATLAB estas líneas deben ser precedidas por el símbolo de porcentaje (%). Se recomienda añadir comentarios al inicio de cada función, en estos se debe explicar el objetivo de la función y los parámetros de entrada. Otra buena práctica es añadir comentarios dentro del código en donde se explique las acciones que se están realizando: definir variables, lectura de datos, actualización de la interfaz, entre otros. Finalmente, los comentarios no deben ser excesivos, su objetivo no es explicar a detalle cada instrucción realizada sino dar una explicación general para una persona que no haya estado involucrada en la codificación. Un código bien redactado debería ser capaz de explicarse por sí solo.

Por último, en lo que respecta la redacción del código, se recomienda implementar un *texto* legible. En este sentido, se debe dar el correcto

espaciamiento entre los nombres de las variables y las operaciones. Evitar redactar más de una instrucción por línea de código. Cuando una instrucción es demasiado larga, 80 caracteres según Johnsons (2002), se recomienda dividir la línea, para MATLAB la división se realiza incluyendo tres puntos (...) entre las partes de la instrucción. El quiebre debe realizarse en una posición que no perjudique el entendimiento de la instrucción y no de manera aleatoria.

Se han presentado solo las recomendaciones más importantes y que han sido tomadas en cuenta en el programa. De esta manera se da opción de que un futuro proyecto pueda ampliar el alcance del programa desarrollado y se pueda crear una herramienta más completa.



CAPÍTULO 6: IMPLEMENTACIÓN

En el presente capítulo se detalla la etapa de implementación del programa. Luego de completar el diseño de las interfaces se traduce las funciones definidas a código. Si bien el diseño se realizó considerando que el programa sería desarrollado en MATLAB, un correcto diseño podría ser implementado en cualquier lenguaje de programación. El lenguaje de programación se escoge en base a los requerimientos del programa, las ventajas y desventajas del lenguaje y los conocimientos del programador.

El código implementado debe respetar los estándares definidos en el Capítulo 5. Luego de completar el código es necesario realizar pruebas para verificar que el programa responde a los requisitos definidos en el Capítulo 2. Estas pruebas son distintas a las que se realizará una vez completado todo el programa y que se presentan en el Capítulo 7. Las pruebas en esta etapa son ejecutadas por el desarrollador mientras que en el periodo de prueba el usuario es un voluntario ajeno al proyecto.

A continuación, se presenta el lenguaje de programación escogido para la construcción/implementación del programa y el entorno utilizado. Luego se resumen los resultados obtenidos en las pruebas realizadas al programa.

6.1. Construcción

Para la construcción del programa se optó por utilizar el modelo de arquitectura de software más sencillo. Por otro lado, para la elección del lenguaje de programación se analizaron dos alternativas en base a la experiencia del desarrollador. Finalmente se optó por el lenguaje más completo y sencillo de utilizar.

6.1.1. Modelo de arquitectura

Actualmente, los programas modernos son capaces de trabajar con elementos externos como bases de datos y servidores. La posibilidad de conectarse a internet y acceder a información en línea es otra de las funcionalidades que poseen los últimos programas. Algunos incluso permiten que varios usuarios a la vez modifiquen un archivo y no necesitan instalarse en la computadora. Estos grandes avances han sido posibles gracias al increíble desarrollo de la ingeniería de software en las últimas décadas.

A pesar de la existencia de este tipo de programas, el alcance del presente proyecto es bastante limitado con respecto a la implementación del programa. Por ello, la arquitectura del programa es bastante simple y se limita a la computadora del usuario. El programa desarrollado fue concebido en base a un modelo "Built-in". Esto quiere decir que el programa no necesita de elementos externos para funcionar correctamente y se ejecuta en la computadora del usuario. El usuario interactúa directamente con la interfaz. Luego, las funciones desencadenadas por los eventos manipulan la información directamente en el disco duro del equipo.

6.1.2. Lenguaje de programación

Desde la etapa de concepción del proyecto, se eligió el lenguaje MATLAB para la implementación del programa. La otra alternativa analizada fue el lenguaje C++. La razón por la cual sólo se consideró estos lenguajes fue la experiencia y conocimientos del estudiante desarrollador del proyecto. Otros lenguajes populares en el medio y que pudieron haber sido una buena alternativa para el desarrollo son Java, Python y .NET.

MATLAB es un lenguaje de programación de alto nivel diseñado para el cálculo numérico, la visualización y el desarrollo de aplicaciones. Cuenta con un entorno interactivo y una amplia librería de funciones matemáticas para álgebra lineal, resolución de ecuaciones diferenciales, análisis de señales, entre otros (MATLAB, 2015).

La característica más resaltante de MATLAB es que todas las variables son definidas como matrices. Por ejemplo, un valor escalar es considerado como una matriz de tamaño 1x1. Una matriz fila o columna es tratada como un vector y una cadena de texto es una matriz de 1xN donde N es la longitud de la cadena. Por defecto todas las variables son matrices y para acceder a uno de los elementos se debe indicar el índice del elemento deseado. Es importante aclarar que el término matriz no limita la cantidad de dimensiones a dos, las variables pueden ser multidimensionales.

La ventaja más grande de MATLAB es su capacidad de procesamiento. MATLAB es un lenguaje enfocado al cálculo numérico, por ello cuenta con una amplia variedad de funciones matemáticas eficientes para el tratamiento de datos. Dada la naturaleza del proyecto, esta ventaja no es

aprovechada ya que el programa únicamente permite ingresar, almacenar y visualizar los datos, los cuales son, en su mayoría, cualitativos. Sólo se realizan cálculos estadísticos sencillos en algunas de las herramientas.

Otra de las ventajas de MATLAB, y motivo por el cual se decidió utilizarlo en lugar del lenguaje de programación C++, es la capacidad de crear interfaces gráficas de usuario (GUI). Una interfaz facilita la interacción entre el usuario y el programa. El usuario puede utilizar las funcionalidades del programa a través de elementos como botones y cuadros de texto.

Finalmente, MATLAB cuenta con su propio entorno de desarrollo, el cual se muestra en la Figura 6.1. Esto quiere decir que no es necesario usar otra aplicación para crear los códigos, a diferencia de C++ que necesita, además de un editor, también un compilador. Tanto el editor como el compilador están integradas en la pantalla principal de MATLAB. Desde la ventana “Editor” se puede redactar, compilar y ejecutar el código. Utilizando la opción “Run” en el menú de herramientas se compila y ejecuta el código.

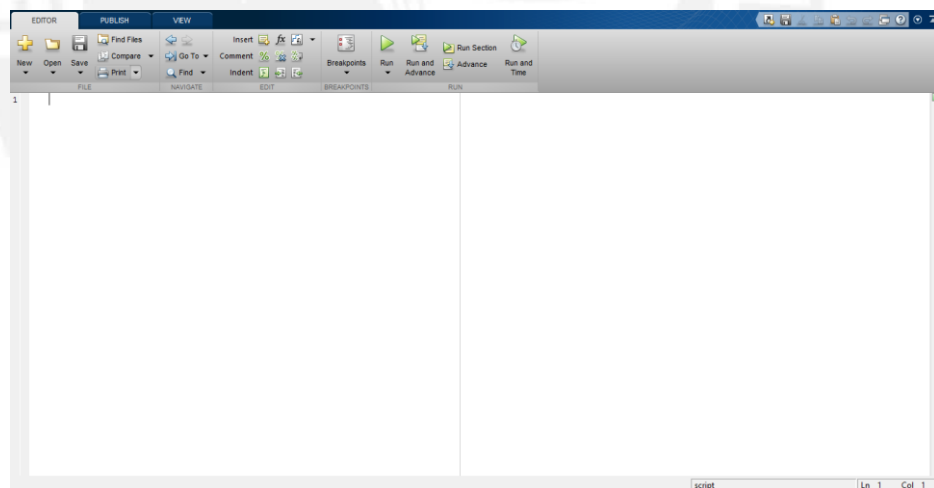


Figura 6.1. Interfaz de la ventana Editor de MATLAB
Fuente: MATLAB 2015a

La única desventaja identificada de elegir el lenguaje de programación MATLAB es que, para ejecutar las aplicaciones, se necesita que el usuario cuente con compilador de MATLAB. Este compilador viene integrado en el programa completo cuyo costo asciende a varios miles de dólares. Un compilador independiente también es distribuido por MATLAB de manera gratuita. Las instrucciones para la ejecución del

programa, así como para instalar el compilador, se encuentran en el anexo de guía de usuario.

6.2. Pruebas Unitarias

Para la ejecución de las pruebas se ha considerado como estrategia utilizar pruebas unitarias, también denominadas pruebas de caja blanca. Una prueba unitaria prueba solo una funcionalidad del programa. Con estas pruebas se busca validar las funcionalidades del programa que atienden a los requerimientos del usuario. A continuación, se resume los resultados de las pruebas realizadas en cada interfaz. Éstas se resumen en tablas donde se indica el código asignado a la prueba, la funcionalidad probada y el resultado obtenido.

6.2.1. Pantalla Principal

Para la pantalla principal fue necesario probar las funcionalidades para crear un proyecto, guardarlo y exportarlo. También se probó el abrir un proyecto guardado e importar un proyecto antes exportado. Finalmente se probó crear y abrir un archivo de cada una de las herramientas implementadas. El código para las pruebas es P-PP-XX que significa Prueba Pantalla Principal número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.1.

Tabla 6.1. Resumen de resultados de pruebas unitarias – Pantalla Principal

P-PP-01: Crear un nuevo proyecto de nombre "Proyecto 1"
Se crea una carpeta de nombre "Proyecto 1" dentro del directorio "Proyectos" en la carpeta principal del programa. Se crea una carpeta por cada una de las herramientas implementadas (Mapping, Tracing, Tracking, Counting, Speed Test y Diary) y una carpeta de nombre "Imágenes".
P-PP-02: Crear un proyecto de nombre "Proyecto 1" cuando ya existe un proyecto del mismo nombre
El programa muestra un mensaje de advertencia de la existencia de un proyecto con el mismo nombre ingresado. Se da la opción al usuario de ingresar un nuevo nombre o de borrar el proyecto existente.
P-PP-03: Guardar el proyecto abierto
Se crea un archivo en formato FIG dentro de la carpeta del proyecto de nombre "Pantalla Principal".
P-PP-04: Abrir un proyecto existente
El programa muestra una relación de las carpetas existentes dentro del directorio "Proyectos".
P-PP-05: Seleccionar un proyecto para abrir
La pantalla actual se cierra y se abre la pantalla guardada del proyecto seleccionado. Todos los elementos están deshabilitados excepto el botón "Actualizar archivos".

P-PP-06: Exportar un proyecto
El programa crea un archivo en formato ZIP dentro del directorio "Exportados" en la carpeta principal del programa. El nombre del archivo coincide con el nombre del proyecto.
P-PP-07: Importar un proyecto
El programa muestra la lista de archivos dentro del directorio "Exportados".
P-PP-08: Seleccionar un proyecto para importar
Se crea una carpeta dentro del directorio "Proyectos" que contiene la información del proyecto importado.
P-PP-09: Crear un archivo utilizado una de las seis herramientas
El programa consulta el nombre para el nuevo archivo. El programa guarda el proyecto y cierra la ventana principal. Inmediatamente se abre la ventana de la herramienta elegida.
P-PP-10: Abrir un archivo existente de una herramienta
El programa cierra la ventana de la pantalla principal y abre la pantalla guardada del registro seleccionado.

Fuente: Elaboración Propia

6.2.2. Herramienta *Tracing*

Para la interfaz de la herramienta *Tracing* fue necesario probar las funcionalidades agregar notas y una imagen de croquis. Una vez cargada la imagen del croquis, se probaron las funciones para crear un trazo, luego se intentó modificarlo, duplicarlo y eliminarlo. Luego de registrar varios trazos con distintas configuraciones se probaron los filtros para la visualización. Finalmente se guardó el archivo y se generó un reporte en formato PDF. El código para estas pruebas es P-TR-XX que significa Prueba *TRacing* número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.2.

Tabla 6.2. Resumen de resultados de pruebas unitarias – Herramienta *Tracing*

P-TR-01: Ingresar notas de texto
El programa muestra una ventana en donde el usuario puede ingresar texto. Se muestra, si hubiese, el texto ingresado previamente.
P-TR-02: Insertar una imagen de fondo
Se muestra una lista de los archivos existentes en la carpeta "Imágenes" en la carpeta del proyecto. Si el archivo elegido no se encuentra en la carpeta mencionada, el programa crea una copia de la imagen con nombre "Tracing –seguido del nombre del archivo. La imagen se muestra en la pantalla.
P-TR-03: Crear un nuevo Trazo
Se solicita ingresar el nombre del trazo. Por defecto se sugiere el nombre "Trazo x" donde x es el menor número que no se encuentra utilizado. Se habilita el panel Menú de Trazo y se muestra el nombre ingresado.
P-TR-04: Iniciar dibujo de un trazo
Se deshabilitan los elementos de la pantalla. El Axes está habilitado para dibujar el trazo con el mouse.

P-TR-05: Guardar el trazo activo
Se deshabilita el panel de Menú de Trazo y restaura la configuración de color e información. Se vuelve a habilitar el Panel de Trazos. El trazo guardado se muestra en el menú desplegable
P-TR-06: Duplicar un trazo
El programa solicita el nombre del nuevo trazo. Se sugiere el nombre del trazo original seguido de "2". Se habilita el panel Menú de Trazo y se carga la información del trazo original con el nuevo nombre.
P-TR-07: Modificar un trazo
El programa habilita el panel Menú de Trazo y carga la información del trazo seleccionado. Se puede modificar los datos y redibujar el trazo.
P-TR-08: Eliminar un trazo
Se solicita la confirmación del usuario y luego se procede a eliminar el trazo seleccionado. El nombre se retira del menú desplegable.
P-TR-09: Modificar los criterios de visualización
Sobre el croquis definido, el programa muestra solo los trazos que cumplen con las condiciones definidas en el panel Menú.
P-TR-10: Guardar el archivo
El programa genera un archivo en formato FIG dentro de la carpeta "Tracing" en el directorio del proyecto.
P-TR-11: Generar reporte PDF
El programa genera un archivo en formato PDF dentro de la carpeta "PDF" en el directorio del proyecto. El archivo lleva de nombre "Tracing" seguido del nombre del archivo. En el PDF se muestra el nombre del proyecto, el nombre del archivo y el croquis ingresado. Sobre el croquis se muestran los trazos que cumplen las condiciones impuestas en el panel Menú.
P-TR-12: Cerrar la ventana
Se solicita la confirmación del usuario y se da la opción de guardar el archivo antes de continuar. La ventana se cierra y se abre la pantalla principal con todos los elementos deshabilitados excepto el botón Actualizar archivos.

Fuente: Elaboración Propia

6.2.3. Herramienta *Mapping*

Para la interfaz de esta herramienta también se debió probar las funcionalidades para agregar notas y una imagen de croquis. De igual manera para guardar el archivo y cerrar la ventana. Como los códigos utilizados fueron los mismos que en la herramienta anterior ya no se mencionarán en la tabla resumen.

Una vez cargada la imagen del croquis, se probaron las funciones para crear una actividad, luego se probó editarla y eliminarla. Dentro de cada actividad se realizaron las pruebas para personalizar el formato de los marcadores y la inserción/eliminación de puntos. Luego de crear varias actividades con distintas configuraciones se probaron los filtros para la visualización. Finalmente se guardó el archivo y se generó un reporte en formato PDF. El código para estas pruebas es P-MP-XX que significa

Prueba MaPping número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.3.

Tabla 6.3. Resumen de resultados de pruebas unitarias – Herramienta Mapping

P-MP-01: Crear una actividad
Se solicita el nombre de la actividad. Se sugiere el nombre “Actividad x” donde x es el menor número que no se encuentra utilizado. Se habilita el panel Actividad.
P-MP-02: Ubicar puntos
Se deshabilitan los elementos de la pantalla. El Axes se habilita para dibujar puntos sobre la imagen.
P-MP-03: Personalizar puntos
Según las modificaciones del color, tipo y tamaño de marcador se actualiza los puntos visualizados.
P-MP-04: Eliminar puntos
El programa retira los puntos seleccionados.
P-MP-05: Guardar actividad
Se deshabilita el panel Actividad y se muestran todos los puntos en el croquis. La actividad se ha añadido al menú desplegable.
P-MP-06: Modificar los criterios de visualización
Sobre el croquis definido, el programa muestra solo los puntos que cumplen con las condiciones definidas en el panel Menú.
P-MP-07: Generar reporte PDF
El programa genera un archivo en formato PDF dentro de la carpeta “PDF” en el directorio del proyecto. El archivo lleva de nombre “Mapping” seguido del nombre del archivo. En el PDF se muestra el nombre del proyecto, el nombre del archivo y el croquis ingresado. Sobre el croquis se muestran los puntos de las actividades que cumplen las condiciones impuestas en el panel Menú.

Fuente: Elaboración Propia

6.2.4. Herramienta *Tracking*

En esta interfaz se probaron los botones para dibujar la trayectoria de análisis y los registros. Se probó el ingreso y eliminación de los registros, la ubicación de los puntos y la modificación del texto. Finalmente se guardó el archivo y se generó un reporte en formato PDF. El código para estas pruebas es P-TK-XX que significa Prueba TracKing número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.4.

Tabla 6.4. Resumen de resultados de pruebas unitarias – Herramienta Tracking

P-TK-01: Dibujar trazo
Se deshabilita los elementos de la interfaz. Se habilita el elemento Axes para dibujar un trazo con el mouse sobre él.
P-TK-02: Añadir registro
Se crea un nuevo registro. El campo donde se indica el registro activo muestra el número del nuevo registro. El campo de texto se reinicia y se borran los puntos sobre el croquis.
P-TK-03: Ubicar punto
Se deshabilita los elementos de la pantalla. El elemento Axes está habilitado para dibujar un punto en la posición donde se haga clic. Luego de dibujarlo se habilita la pantalla nuevamente.
P-TK-04: Eliminar registro
El programa retira el registro seleccionado. Seguidamente se reasigna la numeración a los registros restantes.
P-TK-05: Terminar edición de registros
Se deshabilita el panel Registros y se muestran todos los puntos sobre el croquis.
P-TK-06: Generar reporte PDF
El programa genera dos archivos en formato PDF dentro de la carpeta "PDF" en el directorio del proyecto. Los archivos llevan de nombre "Tracking -" seguido del nombre del archivo y los sufijos "- croquis" y "-registros". En el primer PDF se muestra el nombre del proyecto, el nombre del archivo y el croquis ingresado con la trayectoria y los puntos asociados a los registros. En el segundo PDF se presenta el texto de todos los registros.

Fuente: Elaboración Propia

6.2.5. Herramienta *Counting*

En esta interfaz se probaron los botones para dibujar la sección de análisis y la inicialización de la pantalla para cargar datos. Dentro de esta pantalla se realizaron las pruebas para importar un archivo de Excel. Se comprobaron los resultados obtenidos en el histograma y la tabla de parámetros estadísticos según distintas combinaciones de criterios de sexo, edad y presencia de discapacidades. Finalmente se guardó el archivo y se generó un reporte en formato PDF. El código para estas pruebas es P-CO-XX que significa Prueba COunting número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.5.

Tabla 6.5. Resumen de resultados de pruebas unitarias – Herramienta Counting

P-CO-01: Dibujar sección de análisis
Se deshabilita los elementos de la interfaz. Se habilita el elemento Axes para dibujar un trazo con el mouse sobre él.
P-CO-02: Inicializar ventana para ingresar datos
Se abre la ventana “Cargar Datos”. Si existen datos ingresados previamente se cargan en la tabla.
P-CO-03: Añadir fila en la tabla
Se añade una fila en blanco al final de tabla. Esta fila puede ser modificada por el usuario usando el teclado.
P-CO-04: Importar archivo Excel
Se solicita seleccionar el archivo. Se cargan los datos que se encuentran en la primera hoja del archivo a la tabla de la ventana “Cargar Datos”.
P-CO-05: Terminar carga de datos
Se cierra la venta “Cargar Datos”. En la ventana de la herramienta se actualiza la información mostrada en el histograma y en la tabla de parámetros estadísticos.
P-CO-06: Validación de la información mostrada
Se realizó cálculos en una hoja de Excel para validar los valores que muestra el programa. Ambos resultados coincidieron.
P-CO-07: Generar reporte PDF
El programa genera un archivo en formato PDF dentro de la carpeta “PDF” en el directorio del proyecto. El archivo lleva de nombre “Counting -” seguido del nombre del archivo. En el PDF se muestra el nombre del proyecto, el nombre del archivo y el croquis ingresado. Sobre el croquis se dibuja la sección de análisis. También se imprime el histograma y la tabla de parámetros estadísticos.

Fuente: Elaboración Propia

6.2.6. Herramienta *Speed Test*

En esta interfaz se probaron los botones para dibujar las que delimitan el tramo analizado y la inicialización de la pantalla para cargar datos. Similar a la herramienta anterior, se comprobaron los resultados obtenidos en el histograma y la tabla de parámetros estadísticos. También se verificó que los resultados mostrados en ambas unidades disponibles sean correctos. Finalmente se guardó el archivo y se generó un reporte en formato PDF. El código para estas pruebas es P-ST-XX que significa Prueba Speed Test número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.6.

Tabla 6.6. Resumen de resultados de pruebas unitarias – Herramienta Speed Test.

P-ST-01: Dibujar sección de análisis
Se deshabilita los elementos de la interfaz. Se habilita el elemento Axes para dibujar dos trazos con el mouse sobre él.
P-ST-02: Inicializar ventana para ingresar datos
Se abre la ventana “Cargar Datos”. Si existen datos ingresados previamente se cargan en la tabla.
P-ST-03: Añadir fila en la tabla
Se añade una fila en blanco al final de tabla. Esta fila puede ser modificada por el usuario usando el teclado.
P-ST-04: Importar archivo Excel
Se solicita seleccionar el archivo. Se cargan los datos que se encuentran en la primera hoja del archivo a la tabla de la ventana “Cargar Datos”.
P-ST-05: Terminar carga de datos
Se cierra la venta “Cargar Datos”. En la ventana de la herramienta se actualiza la información mostrada en el histograma y en la tabla de parámetros estadísticos.
P-ST-06: Validación de la información mostrada
Se realizó cálculos en una hoja de Excel para validar los valores que muestra el programa. Ambos resultados coincidieron. También se verificaron los resultados luego de modificar las unidades.
P-ST-07: Generar reporte PDF
El programa genera un archivo en formato PDF dentro de la carpeta “PDF” en el directorio del proyecto. El archivo lleva de nombre “Speed Test -” seguido del nombre del archivo. En el PDF se muestra el nombre del proyecto, el nombre del archivo y el croquis ingresado. Sobre el croquis se dibuja la sección de análisis. También se imprime el histograma y la tabla de parámetros estadísticos.

Fuente: Elaboración Propia

6.2.7. Herramienta *Diary*

En esta interfaz se probó el botón para crear un nuevo registro y para añadir imágenes. También se probó el slider que itera sobre los registros existentes. Finalmente se guardó el archivo y se generó un reporte en formato PDF. El código para estas pruebas es P-DI-XX que significa Prueba Diary número XX. El resumen de las pruebas se presenta a continuación en la Tabla 6.7.

Tabla 6.7. Resumen de resultados de pruebas unitarias – Herramienta Diary.

P-DI-01: Crear nuevo registro
Se crea un nuevo registro. El campo donde se indica el registro activo muestra el número del nuevo registro. El campo de texto se reinicia y se borra la imagen en el Axes.
P-DI-02: Importar imagen
Se solicita al usuario seleccionar la imagen. Se borra la imagen que se muestra en el Axes y se dibuja la nueva.
P-DI-03: Iterar en los registros existentes
Se al cambiar a otro registro se actualiza la imagen en el Axes, el texto mostrado y la hora y fecha.
P-DI-04: Generar reporte PDF
El programa genera archivos en formato PDF dentro de la carpeta “PDF” en el directorio del proyecto. Los archivos llevan de nombre “Diary -” seguido del nombre del archivo. En cada PDF se muestra el nombre del proyecto, el nombre del archivo y tres registros con su respectiva fotografía. La cantidad de archivos depende de la cantidad de registros.

Fuente: Elaboración Propia

Todas las pruebas realizadas fueron satisfactorias y permitieron validar, en un primer nivel, el programa desarrollado. Llegado a este punto el programa está completado y puede pasar por un periodo de pruebas más riguroso a cargo de usuarios ajenos al proyecto y con datos recolectados en un verdadero estudio de vida pública. Esta validación se presenta en el siguiente capítulo.

CAPÍTULO 7: PERIODO DE PRUEBAS

En el presente capítulo se presenta un resumen de las pruebas realizadas con el programa. A diferencia de las pruebas unitarias que se presentaron en el Capítulo 6, estas pruebas fueron realizadas por personas externas al proyecto y con datos de estudios realizados. De esta manera se pone a prueba el programa en un entorno real. Para realización se solicitó la colaboración de personas externas.

De las pruebas se obtuvo sugerencias y apreciaciones por parte de los usuarios mientras que otros detalles fueron identificados por el desarrollador. Según la información recolectada se realizaron mejoras en el programa; sin embargo, algunas de las recomendaciones solicitaban mejoras que escapan a los alcances del proyecto o que, por las limitaciones de MATLAB, resultan inviables.

7.1. Proyecto Estudio de vida pública y microsimulación en la calle Marqués de la ciudad de Cusco

El primer colaborador fue el estudiante de ingeniería civil Jorge Luis Zúñiga Castañeda. Él aceptó probar el programa en el desarrollo de su proyecto de fin de carrera en el semestre 2016 – 1. Como parte del mencionado proyecto se desarrolló un estudio de vida pública en la calle Marqués ubicada en la ciudad de Cusco (Zúñiga, 2017).

Las pruebas fueron realizadas del 25 al 28 de mayo del 2016. El uso del programa fue supervisado por el desarrollador para identificar los errores que se presentaron. Al finalizar las pruebas se recopiló la apreciación del usuario y sus recomendaciones para mejorar el programa. A continuación, se presentan capturas de pantalla recopiladas durante el periodo de prueba. Cada captura corresponde a cada herramienta empleada.

Al utilizar la herramienta de mapeo, Figura 7.1, no se encontró mayores inconvenientes con respecto a la funcionalidad del programa. Sin embargo, se puede apreciar que cuando la cantidad de actividades es mayor a diez, la leyenda invade el espacio del croquis. Se reubicó el elemento Axes para reservar un espacio a la leyenda. Esta nueva distribución también se tomó en cuenta al generar el reporte.

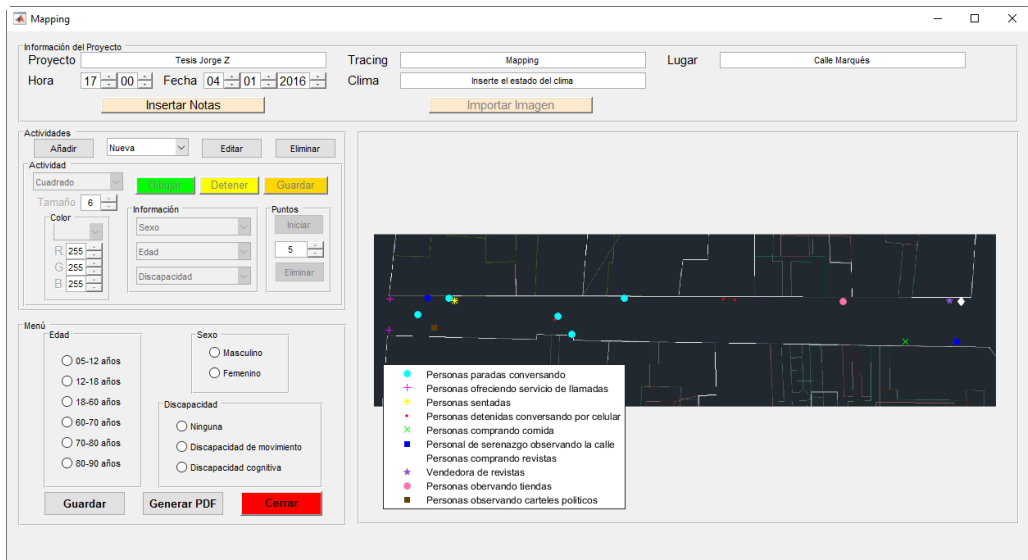


Figura 7.1. Interfaz herramienta Mapping
Fuente: Usuario de prueba 1 – Jorga Zúñiga

En la herramienta para el trazado de rutas, Figura 7.2, se sugirió disminuir el espesor las líneas ya que estas se aglomeran y se confunden entre sí. Se detectó que la forma de dibujar los trazos, arrastrando el cursor en la pantalla, resulta complicado para trazar tramos rectos. Se sugirió, en una futura ampliación del programa, permitir dibujar los trazos rectos señalando los extremos de los segmentos que lo componen.

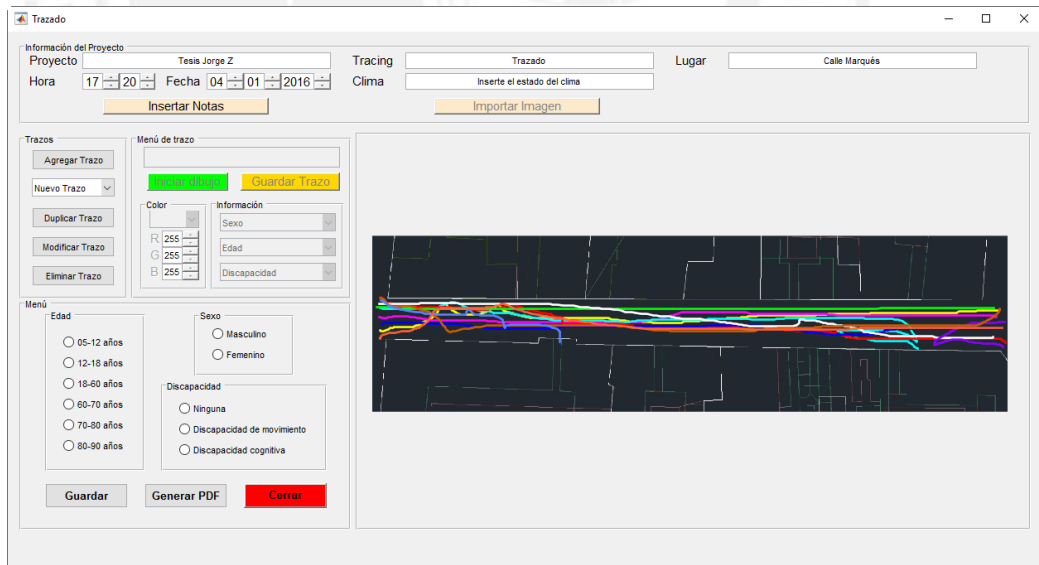


Figura 7.2. Interfaz herramienta Tracing
Fuente: Usuario de prueba 1 – Jorga Zúñiga

En la herramienta de seguimiento, Figura 7.3, no se identificaron inconvenientes excepto por el dibujo de trazos rectos que se mencionó anteriormente.

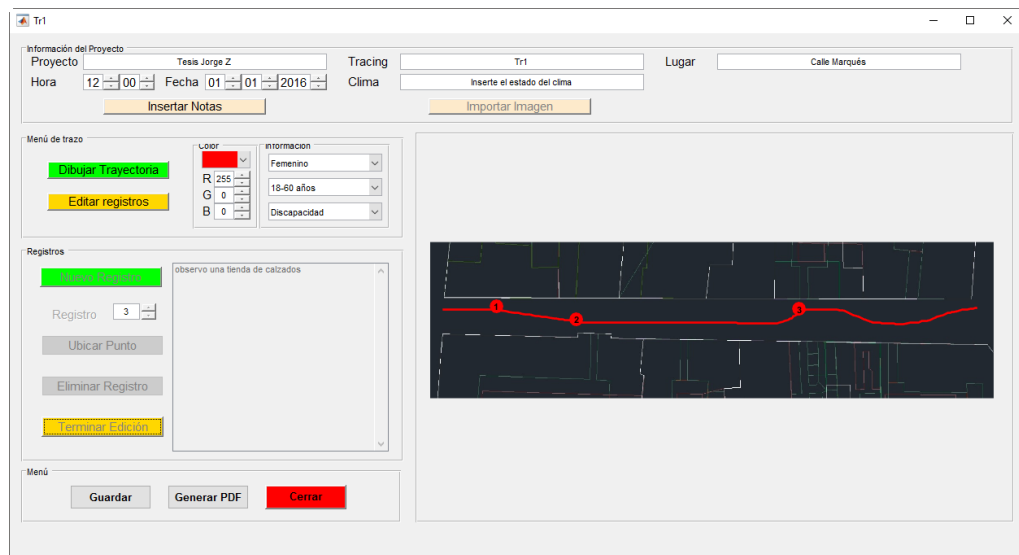


Figura 7.3. Interfaz herramienta Tracking
Fuente: Usuario de prueba 1 – Jorga Zúñiga

Durante la ejecución de la herramienta conteo, Figura 7.4, no se llegó a identificar errores. En el estudio realizado solo se realizó un registro, por ello en el histograma solo se aprecia una barra.

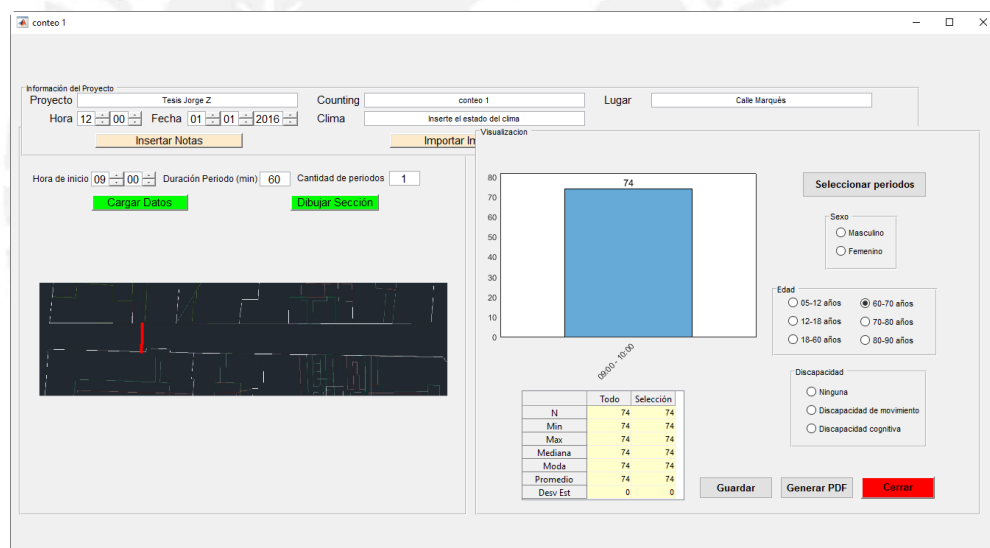


Figura 7.4. Interfaz herramienta Counting
Fuente: Usuario de prueba 1 – Jorga Zúñiga

Al usar la herramienta Speed Test si se pudo probar la funcionalidad para cargar archivos de Excel. Se sugirió especificar, y de ser posible mostrar un ejemplo, del formato que el archivo Excel debe tener para aclarar el correcto uso del programa para el usuario. Dicha información fue especificada en la guía de usuario.

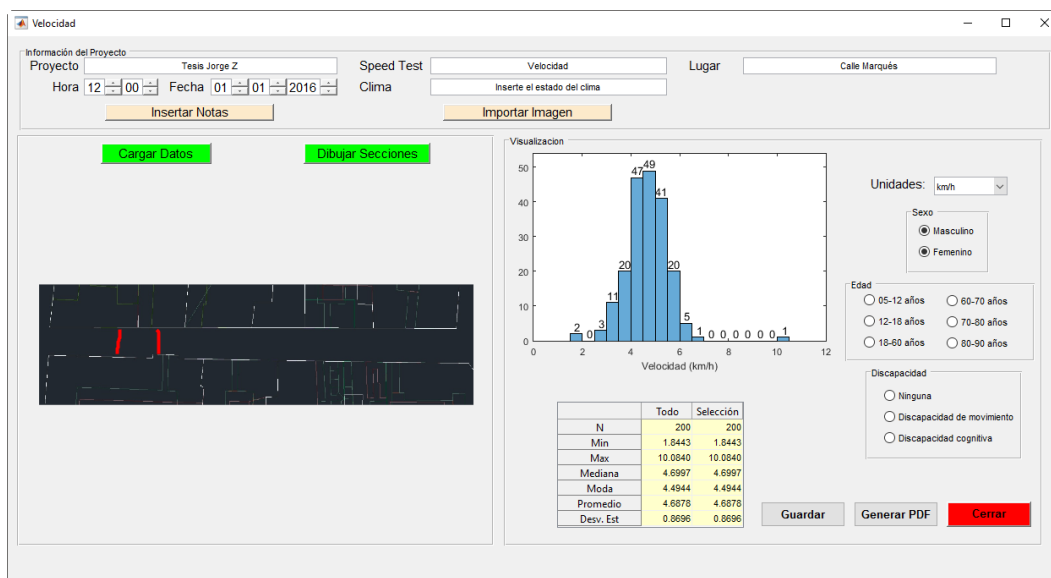


Figura 7.5. Interfaz herramienta Speed Test
Fuente: Usuario de prueba 1 – Jorga Zúñiga

Finalmente, para herramienta Diary, Figura 7.6, no se identificó ningún inconveniente en términos de funcionalidad del programa. Sin embargo, se pudo apreciar que, dependiendo del tamaño de las fotografías, el archivo obtenido puede resultar muy pesado (mayor a 50 MB). Esto ocurre cuando las imágenes cargadas son muy grandes. Por ello, se optó por guardar, en lugar de la imagen dentro del archivo, una copia de la imagen dentro de la carpeta del proyecto y solo almacenar la ruta del archivo seleccionado. De esta manera bastará cargar cada imagen cada vez que se desee mostrarla. Esto puede llegar a resultar pesado en tiempo de ejecución, pero reduce el tamaño del archivo.

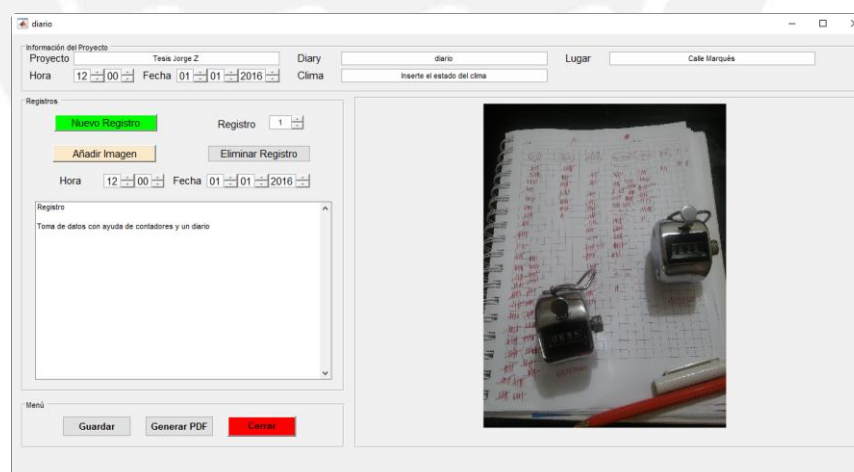


Figura 7.6. Interfaz herramienta Diary
Fuente: Usuario de prueba 1 – Jorga Zúñiga

En general, el usuario resulto satisfecho con el programa y realizó algunos comentarios generales con respecto al aspecto estético del programa. Se sugirió ampliar la cantidad de caracteres disponibles para nombrar los archivos e incluir algunos como el guion (-), coma (,), y punto (.). Sin embargo, la decisión de solo

permitir caracteres alfanuméricos se tomó para evitar ambigüedades en el sistema. Es posible que al crear elementos o archivos con estos caracteres se produzca un error en el sistema operativo. Se optó por añadir el carácter (_). Por otro lado, también se solicitó que los reportes en formato PDF sean abiertos luego de ser creados, esta funcionalidad fue añadida a la versión final del programa.

7.2. Proyecto Estudio de vida pública y microsimulación peatonal en la vía PUCP

El presente proyecto de tesis pertenece a los estudiantes Gabriel Sergio Luis Legua Landeo y Manuel German Sabino Gonzales. La información perteneciente al proyecto fue brindada por el asesor Félix Cabrera Vega con el objetivo de probar las funcionalidades del programa. El estudio fue realizado en la vía peatonal principal del campus PUCP, comúnmente llamada Tontódromo. El espacio analizado se delimita por la Biblioteca Centra y la Biblioteca de Ciencias e Ingeniería.

En estas pruebas, el programa no fue utilizado por personas externas sino por el desarrollador. A continuación se presenta un resumen de los resultados obtenidos. Con los datos provistos se utilizaron cuatro herramientas.

En la Figura 7.7 se observa la interfaz de la herramienta Mapping. Al igual que con la prueba anterior, se notó que el espacio delimitado para la leyenda no es suficiente cuando se tienen múltiples actividades. Otra observación fue que el programa está diseñado para validar que los nombres ingresados sean cadenas de texto alfanuméricas, pero no se incluía las vocales tildadas. Se corrigió este error para permitir los caracteres mencionados.

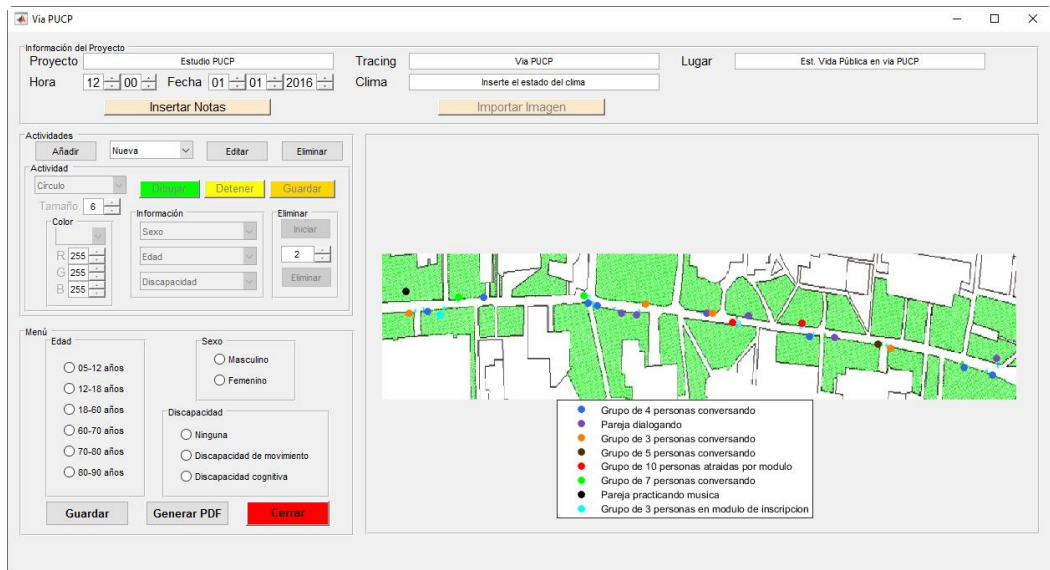


Figura 7.7. Interfaz herramienta Mapping
Fuente: Prueba 2 – José Mallma

En la Figura 7.8 se observa la herramienta Counting y en la Figura 7.9 la herramienta Speed Test. En general no se presentaron inconvenientes al utilizar ambas herramientas. Un error menor fue identificado: la herramienta Speed Test guardaba los archivos en la carpeta de la herramienta Counting.

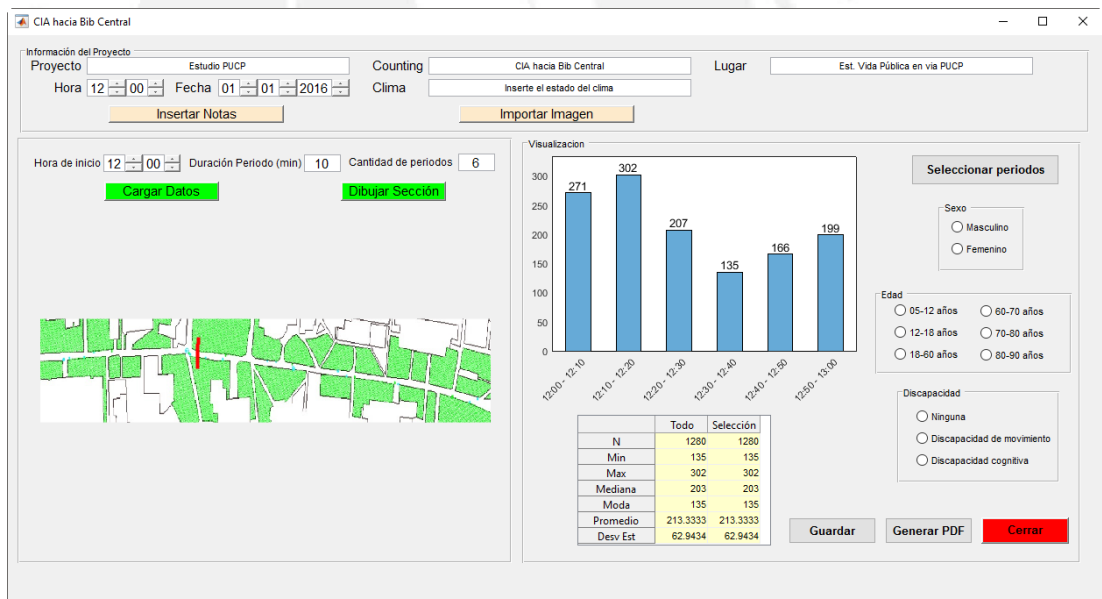


Figura 7.8. Interfaz herramienta Counting
Fuente: Prueba 2 – José Mallma

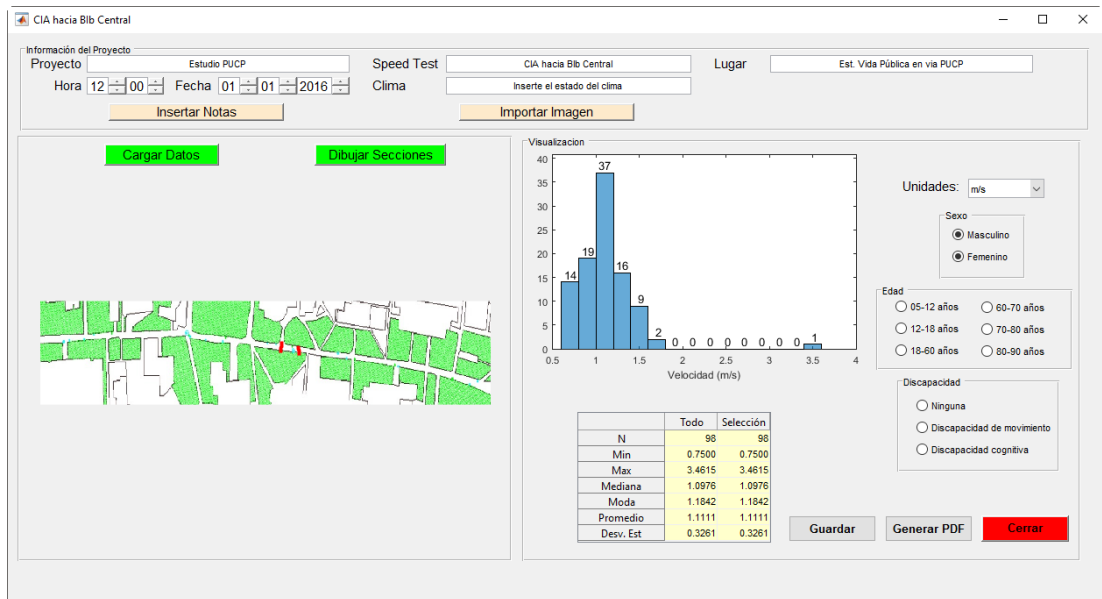


Figura 7.9. Interfaz herramienta Speed Test
Fuente: Prueba 2 – José Mallma

Finalmente, se probó la herramienta Diary, la cual se muestra en la Figura 7.10. No se presentaron mayores inconvenientes; sin embargo, se notó que el espacio brindado para ingresar la nota de texto es mayor al permitido al momento de generar el reporte. Esto significa que, si el usuario llenara todo el campo disponible, la mitad del texto ingresado no sería visible en el reporte PDF. Por ello se optó por incrementar el tamaño del texto de manera que el usuario sepa el límite de texto que puede ingresar en cada registro.

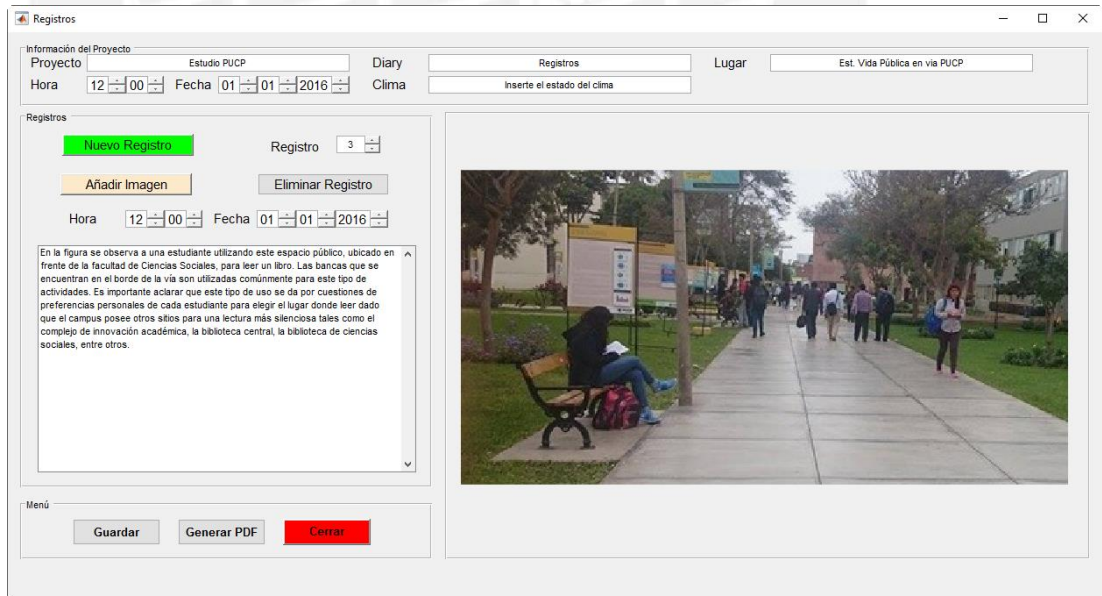


Figura 7.10. Interfaz herramienta Diary
Fuente: Prueba 2 – José Mallma

CAPÍTULO 8: CONCLUSIONES Y RECOMENDACIONES

El presente capítulo presenta las recomendaciones y conclusiones del desarrollo del proyecto. Se espera que éstas puedan servir como base para futuros proyectos. Por un lado, que sirva como herramienta para estudios de vida pública y a futuro se profile como una forma de estandarizar la información recolectada. Por otro lado, que sirva como experiencia para futuros proyectos en la línea del desarrollo de aplicaciones y programación en MATLAB.

8.1. Conclusiones

Con base en las experiencias de distintas ciudades, la planificación de la ciudad debe seguir los lineamientos de la movilidad sostenible. Bajo esta premisa, el objetivo es diseñar la ciudad, y por tanto el espacio público, tomando en consideración las necesidades de todos los usuarios.

Aunque para muchos pueda resultar incorrecto, las necesidades a tomar en cuenta no se limitan al ámbito del desplazamiento. El espacio público también cumple un rol social que no debe ser omitido. Por ello, los usos que se le da al espacio público dependen de muchos factores como el entorno físico: clima u hora; las características de las personas: edad, sexo o la presencia de discapacidades; entre otros.

Dada la complejidad del comportamiento humano en el espacio público, o también llamado Vida Pública, distintos autores proponen la observación como principal herramienta para su estudio. La observación permite identificar no solo las acciones, sino también sus causas y consecuencias. Con los datos recolectados se puede diseñar un espacio acorde a las necesidades de la población y que no limite sus acciones en ningún aspecto.

Gehl y Svarre (2013) proponen diversas metodologías para recopilar datos de la vida pública. Con base en dichas metodologías, o herramientas, se planteó el desarrollo de una herramienta informática para estandarizar el almacenamiento y visualización de los datos recolectados.

Para el desarrollo del programa se optó por la metodología propuesta por el modelo incremental. Siguiendo este lineamiento, se aprovechó al máximo la curva de aprendizaje. El tiempo de implementación de cada incremento se redujo a medida que se avanzaba con las herramientas. Múltiples funciones fueron

reusadas en distintas instancias mientras que otras eran optimizadas según se avanzaba en el proyecto.

El tiempo y esfuerzo invertido en las etapas de análisis y diseño permitió identificar y atender los requerimientos más importantes. Por ello, no fue necesario realizar reprocesos mayores ni descartar avances. Si bien existieron modificaciones menores, estas no demandaron más de un día de trabajo por lo que el cronograma no se vio comprometido.

No fueron necesarios conocimientos avanzados en algoritmos o estructuras de datos complejas para el desarrollo del programa. La codificación de todas las herramientas se realizó con base en estructuras de control sencillas como iterativas y selectivas.

Para la implementación del programa se utilizó el lenguaje de programación MATLAB. La ventaja de esta plataforma radica en dos aspectos. En primer lugar, el programa cuenta con su propio entorno de desarrollo, por lo que no es necesario utilizar otro programa, y con una amplia librería de funciones nativas. En segundo lugar, la herramienta GUIDE permite diseñar interfaces de usuario de calidad y que facilitan la interacción con el usuario. Una tercera ventaja del lenguaje, que no fue aprovechada por la naturaleza del proyecto, pero que es importante mencionar es la eficiencia de ejecución. Esto quiere decir que cuando se realizan cálculos complejos en grandes bases de datos no se consumen recursos de tiempo y memoria como en otros lenguajes de programación.

Con base en las pruebas realizadas, si bien el programa es una herramienta básica en comparación a softwares profesionales, se cumple con las expectativas planteadas al inicio del proyecto. Las pruebas también permitieron identificar funcionalidades que podrían ser implementadas en una futura ampliación del proyecto ya que escapan al alcance del presente proyecto.

Finalmente, es importante señalar que, a la fecha, no existen antecedentes de un proyecto similar. Los estudios de vida pública recolectan información en su mayoría cualitativa por lo que cada registro varíe según la persona que realizó el estudio. Esta heterogeneidad dificulta realizar un análisis con base en distintas fuentes. El principal objetivo del proyecto es que la herramienta desarrollada plantee, a futuro, un formato estándar para el registro de los datos y facilite su interpretación. De la información obtenida es posible desarrollar modelos de

comportamiento peatonal para su simulación y diseñar espacios públicos más accesibles e incluyentes.

8.2. Recomendaciones

Dada la naturaleza del proyecto, las recomendaciones que se plantearán están enfocadas al desarrollo de herramientas informáticas y la programación. Previamente, es necesario señalar que, a la fecha, no existe una línea en la espacialidad de ingeniería civil en donde se brinde conocimientos en programación. Existe, en el primer ciclo para todas las especialidades, un curso llamado Introducción a la Computación en donde se da un primer acercamiento a la programación. Sin embargo; para los estudiantes de ingeniería civil es el único contacto directo con el tema.

En el mundo actual las herramientas informáticas han cobrado gran importancia, más aún en un rubro como la ingeniería. No existe un ingeniero civil, sin importar la rama a la que se dedique, que no necesite utilizar softwares. Sin embargo, muchas veces es necesario realizar operaciones sencillas pero tediosas que requieren de un programa al cual no se tiene acceso o simplemente no existe un programa que realice esa tarea en particular. En este punto es donde nace la necesidad de desarrollar programas propios para satisfacer los requerimientos. No es necesario contar con programas muy especializados ya con programas en lenguajes como C++ y Python, o macros en Excel (VBA) se pueden obtener múltiples resultados. Por ello es importante que el ingeniero cuente con los conocimientos básicos en programación y que sea capaz de, en base a ellos, elaborar soluciones más complejas.

Desde un punto de vista personal y con base en la experiencia del estudiante, se pueden definir dos niveles de programación. Un primer nivel básico en el cual se dan instrucciones simples a la máquina como sumar dos números o mostrar una palabra en la pantalla. El segundo nivel, en donde realmente se mide la habilidad del programador y se aprovecha el potencial de una computadora, involucra soluciones mucho más complejas pero eficientes. La capacidad de abstraer un problema en particular y diseñar una manera eficiente de resolverlo es una habilidad tan valiosa como saber redactar un texto formal o saber diseñar una sección de concreto armado. Desarrollar tales habilidades incrementar el valor del profesional en el mercado laboral.

Por ello, se recomienda fomentar las habilidades de programación, ya sea mediante talleres o incluso cursos dedicados al desarrollo de algoritmos e implementación de estructuras de datos.

Pasando al ámbito de la herramienta desarrollada, de lo aprendido se resalta la importancia de las etapas de análisis y diseño en el ciclo de vida de un proyecto de software. Una correcta identificación de los requerimientos y un adecuado diseño agilizan la implementación del programa. Al momento de redactar el código, se recomienda seguir las denominadas buenas prácticas de programación. De esta manera se facilita el entendimiento del código para su mantenimiento. Además, se obtiene un código más limpio y ordenado.

Para el desarrollo de aplicaciones se recomienda integrar el programa con una interfaz gráfica de usuario. La interfaz facilita la interacción entre el usuario y el programa. De esta forma no es necesario que el usuario tenga conocimientos de programación.

En lo que respecta al desarrollo de futuros proyectos de ampliación del programa desarrollado se recomienda integrar métodos más eficientes de visualización y generación de reportes. MATLAB cuenta con herramientas adicionales que permiten obtener resultados más elaborados como el Toolbox "Report Generator". También se sugeriría la implementación de la herramienta en otro lenguaje que permita crear aplicaciones más interactivas. Lenguajes como JAVA o Python serían los más recomendables.

Finalmente, se recomienda evaluar el programa en proyectos reales de estudios de vida pública. Según los resultados que se obtengan se puede modificar las herramientas existentes o añadir otras que no hayan sido contempladas.

REFERENCIAS

- Borja, J. (2003). *La ciudad conquistada*. Madrid: Alianza.
- Buchanan, C. (1964). *Traffic in towns*. Harmondsworth, Inglaterra: Penguin Books.
- Dextre, J., & Avellaneda, P. (2014). *Movilidad en zonas urbanas*. Lima: Fondo Editorial de la Pontificia Universidad Católica del Perú.
- Gehl, J. (1968). People on Foot. *Arkitekten*,(20), 429-446.
- Gehl, J., & Svarre, B. (2013). *How to study public life*. Island Press.
- Jacobs, J. (1961). *The Death and Life of Great American Cities*. New York: Random House.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Reading, Mass: Addison-Wesley.
- Jimenez, F. (2013, Junio 17). *2do Conversatorio Vida Urbanas: La problemática del tránsito y la ausencia de una solución social*. Lima: Grupo CIVILIZATE.
- Johnson, R. (2002, Octubre). *MATLAB Programming Style Guidelines*. Retrieved from Datatool: http://www.datatool.com/downloads/matlab_style_guidelines.pdf
- Ludeña, W., García, J., & Zucchetti, A. (2013). *Lima y espacios públicos: perfiles y estadística integrada 2010*. Lima: Pontificia Universidad Católica del Perú, Facultad de Arquitectura y Urbanismo, Oficina de Publicaciones.
- MATLAB. (2016). *MathWorks Inc., MATLAB Documentation, MathWorks*. Retrieved from MATLAB Documentation: <http://www.mathworks.com/help/matlab/index.html>
- MATLAB. (2016). *MATLAB - File Exchange*. Retrieved from www.mathworks.com/matlabcentral/fileexchange
- Ministerio de Vivienda. (2006). *Reglamento Nacional de Edificaciones*. El Peruano.
- Oz Radiano. (2014, Octubre 19). *How to get all files under a specific directory in MATLAB?* Retrieved from Stack Overflow: <http://stackoverflow.com/questions/2652630/how-to-get-all-files-under-a-specific-directory-in-matlab>
- Presidencia de la República de Colombia. (1998, Agosto 4). Decreto 1504. *Decreto de Espacio Público*. Bogotá, Colombia: Diario Oficial No. 43357.
- Project Management Institute. (2013). *Guía de los Fundamentos para la Dirección de Proyectos PMBOK (5ta ed.)*.
- Radelat, G. (2003). *Principios de Ingeniería de tránsito*. Washington D.C.: Instituto de Ingenieros de Transporte.
- Remy, J., & Voyé, L. (1981). *Ville: ordre et violence*. Paris: PUF.
- Sansaw. (2012, Noviembre 6). *Drawing lines by dragging the mouse in matlab*. Retrieved from Stack Overflow: <http://stackoverflow.com/questions/13258133/drawing-lines-by-dragging-the-mouse-in-matlab>

- Sanz, A. (2009). *La movilidad sostenible en la planificación urbanística y territorial*. Pamplona, España: NASURSA, Navarra de Suelo Residencial S.A. y CRANA, Centro de Recursos Ambientales de Navarra.
- Sanz, A. (2010). *Urbanismo y movilidad sostenible. Guía para la construcción de ciudades siguiendo criterios de movilidad sostenible*. Pamplona, España: NASURSA, Navarra de Suelo Residencial S.A. y CRANA, Centro de Recursos Ambientales de Navarra.
- Tonucci, F. (2004). *La ciudad de los niños. Un nuevo modo de pensar la ciudad*. Buenos Aires: Losada.
- Tyler, N. (2002). *Accessibility and the Bus System: From Concepts to Practice*. Londres: Thomas Telford.
- Vega Centeno, P. (2006). *El espacio público La movilidad y la revaloración de la ciudad*. Lima: Departamento de Arquitectura - PUCP.
- William, W. (1980). *The social life of small urban spaces*. Washington, D.C.: Conservation Foundation.
- Zúñiga, J. (2017). *Micro-simulación peatonal y estudios de vida pública en la calle Marqués de la ciudad del Cusco (tesis de pregrado)*. Pontificia Universidad Católica del Perú, Lima, Perú.

