



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Puesto como Ingeniero de Pruebas para
Continental Automotive Guadalajara México**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Germán Ulises Punzo Medina

ASESOR DE INFORME

Dr. Edmundo Gabriel Rocha Cózatl



Glosario

- **Caso de prueba:** es una serie de condiciones bajo las cuales se determina si un sistema cumple con el comportamiento o característica esperado. [1]
- **CMD:** también conocido como *Command Prompt* o símbolo del sistema, es el intérprete de comandos del sistema operativo Windows.
- **Expresión regular:** también conocido *Regex*, es una secuencia de caracteres que conforman un patrón de búsqueda. Suelen ser utilizados para encontrar una determinada combinación de caracteres dentro de una cadena de texto.
- **Flashear:** cargar datos en un chip de memoria de estado sólido, en especial los que contienen el sistema operativo para dispositivos y periféricos electrónicos.
- **Falso negativo:** cuando la ejecución de una prueba detecta un defecto cuando en realidad no lo es. [1].
- **Falso positivo:** cuando la ejecución de una prueba no muestra fallos pese a que existe un error en la aplicación. [1]
- **Hotspot:** punto de acceso que permite conexión a una red local inalámbrica creado desde una computadora o un celular.
- **Jenkins:** servidor multiplataforma y accesible mediante interfaz web que se utiliza para compilar y probar proyectos de *software* de forma continua.
- **Metodologías ágiles:** son estrategias de desarrollo de proyectos enfocados principalmente a *software* que buscan flexibilidad en la planeación, es decir, las tareas se van asignando conforme el proyecto se va desarrollando. La metodología ágil más conocida es Scrum. [2]
- **Red privada:** es una red completamente separada de la red pública que conecta un número controlado de ordenadores, ya que usa un espacio de direcciones IP especificadas. Se suelen usar en oficinas y empresas por seguridad debido a que se restringe el acceso a los ordenadores desde internet.
- **Red privada virtual:** es una tecnología de red de ordenadores que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet. Permite que un ordenador conectado internet establezca conexión a una red privada, con toda la funcionalidad, seguridad y políticas de gestión de dicha red privada.
- **Repositorio:** son archivos donde se almacenan recursos digitales de manera que estos pueden ser accesibles a través de internet. Pueden ser de acceso público o estar protegidos y necesitar de una autenticación previa.
- **Script:** son grupos de código escrito en un lenguaje de programación usados para dar forma y función a herramientas de *software*.
- **Scrum:** es un marco de trabajo en el que se aplican un conjunto de prácticas para trabajar colaborativamente caracterizado por adoptar una estrategia de desarrollo incremental y solapar las diferentes fases de desarrollo en lugar de realizar una tras otras. [2]

- **Timeout:** también conocido como tiempo de espera, es un parámetro relacionado con un evento forzado diseñado para ocurrir al final de un tiempo de transcurrido predeterminado.
- **Unidad compartida:** también conocido como *shared drive*, son espacios de almacenamiento informático donde se pueden guardar archivos y acceder a ellos desde cualquier dispositivo.

Introducción

En el siguiente informe se redactan de manera detallada las actividades desempeñadas como Ingeniero de Pruebas en las Oficinas de Investigación y Desarrollo de Continental, atendiendo tareas orientadas a la comprobación de la calidad en el desarrollo de un proyecto de telemática, específicamente en su módulo de WIFI.

Las responsabilidades principales fueron:

- Análisis de requerimientos del cliente.
- Diseño, implementación y automatización de casos de prueba.
- Ejecución de pruebas y recopilación de resultados.
- Reporte detallado y seguimiento de los errores del sistema encontrados.
- Elaboración y actualización de documentación de las pruebas.

Estas actividades se dirigieron a mejorar la detección de errores de *software* en el proyecto, a través de una exploración más profunda del sistema, así como a reducir el tiempo dedicado por el ingeniero de pruebas en la ejecución de los casos de prueba.

Empresa donde se laboró: Continental Engineering Services

Continental Engineering Services fue fundada en 2006 como una subsidiaria de Continental AG, una empresa de origen alemán con sede principal en Hanover, Alemania. Ésta cuenta con alrededor de 193,000 empleados en 58 países y es la cuarta mayor fabricante de neumáticos en el mundo después de Bridgestone, Michelin y Goodyear.

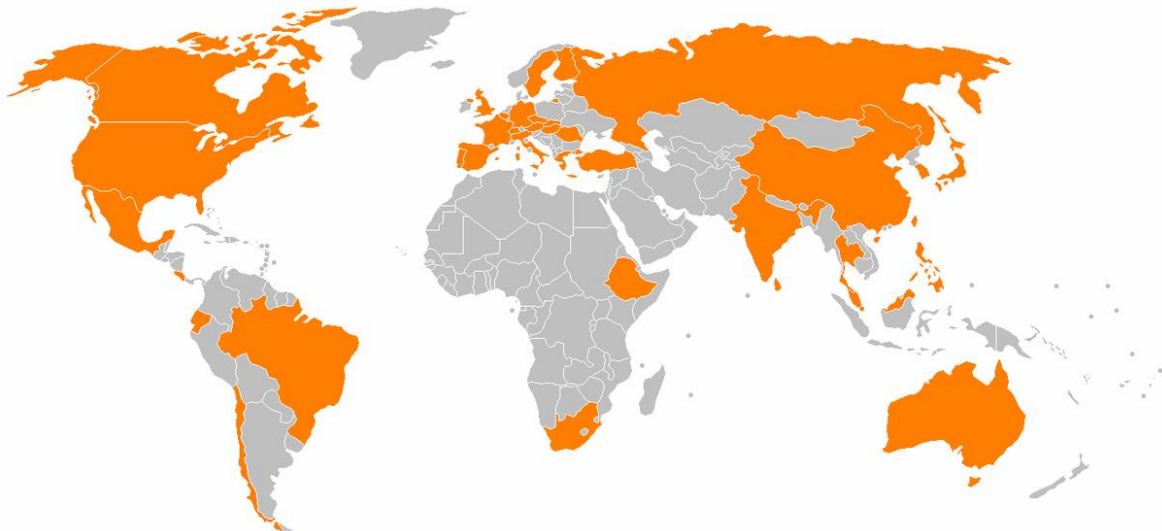


Figura 1: Presencia de Continental en el mundo. [6]

Continental AG se fundó en 1871 como fabricante de caucho bajo el nombre de Continental-Caoutchouc und Gutta-Percha Compagnie en Hanover. La empresa se constituyó desde sus inicios como sociedad anónima cotizando en la bolsa. En 1875 la empresa contaba solo con 250 trabajadores, y tras la era Selignmann, quien fue un directivo destacado hasta 1914, ya habría alrededor de 13,000, volviéndose así líder en los productos de goma y neumáticos en Alemania. En 1892 sería la primera empresa en Alemania en crear neumáticos de aire para bicicletas, en 1898 la primera en crear neumáticos para vehículos sin perfil y en 1904 con perfil. [3]

Por otro lado, Continental Engineering Services se ha convertido en líder en el desarrollo y producción internacional para la industria automotriz. La compañía desarrolla y produce nuevas soluciones para desafíos tecnológicos en los campos de interiores de vehículos, tren motriz y chasis o adapta tecnologías de producción en masa para su uso en series pequeñas, nicho, dos ruedas y vehículos especiales, así como en aplicaciones industriales. [4]

En México, Continental cuenta con 21 locaciones entre las cuales hay fábricas, oficinas, y centros de investigación y desarrollo repartidos en 12 estados de la república. Su planta en Guadalajara, la cual lleva por nombre Continental Automotive Guadalajara México, inició operaciones en el año 1993 y actualmente cuenta con más de 3,700 empleados. [5]

Perfil del puesto y actividades desempeñadas del Ingeniero de Pruebas

La función de los ingenieros de pruebas consiste en corroborar que el producto, componente o sistema se comporta tal y como los requerimientos de dicho proyecto lo solicitan para garantizar la calidad durante todo el ciclo de producción. En otras palabras, los ingenieros de pruebas verifican que el producto funciona de la forma esperada en cada etapa de desarrollo aplicando diferentes métodos de análisis y exploración. Dependiendo el producto, se ejecutan pruebas en diversos componentes y funciones para detectar y reparar problemas mecánicos, eléctricos, electrónicos o de *software*.

En particular, el puesto desempeñado para Continental requiere que el Ingeniero de pruebas sea responsable de garantizar la calidad del sistema en los módulos electrónicos verificando que el *software* cumpla con los requerimientos del cliente; por lo que previamente se requiere analizar dichos requerimientos, el sistema en general y sus componentes para diseñar los casos de prueba adecuados y, finalmente, ejecutar las pruebas necesarias para corroborar el correcto funcionamiento del sistema.

A continuación, se muestra el organigrama para el puesto de Ingeniero de Pruebas en el cual me desempeñé:

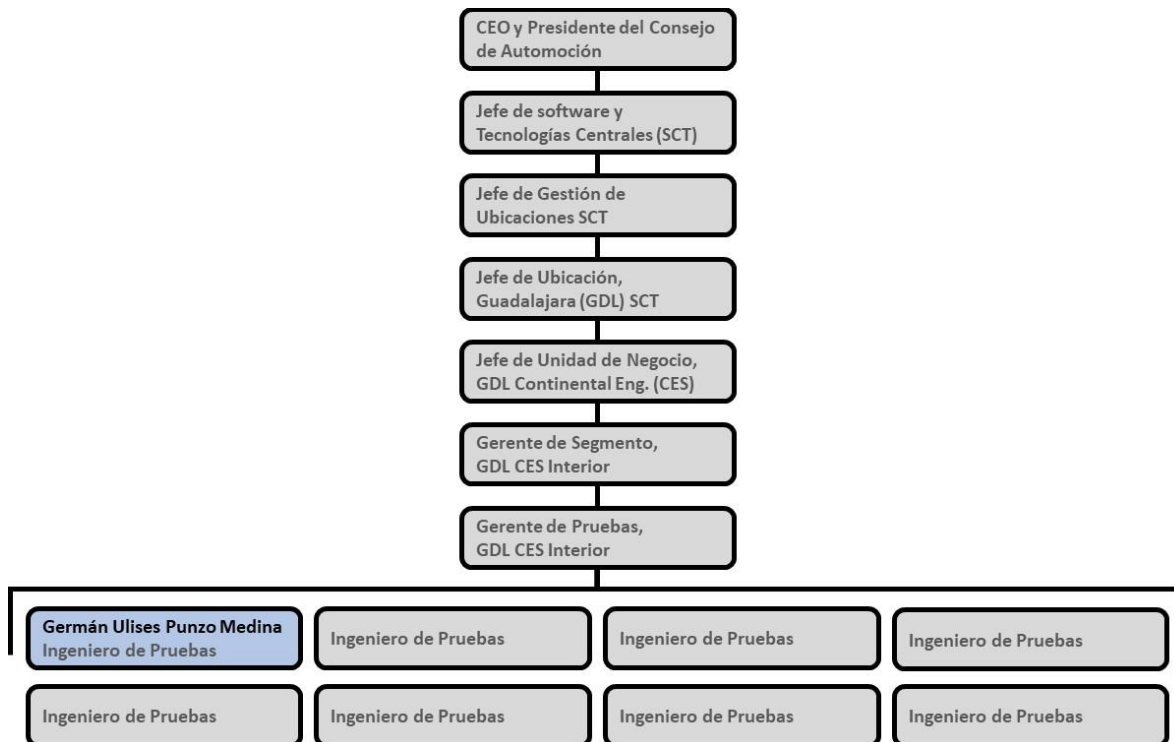


Figura 2: Organigrama de la empresa. [7]

Algunas actividades adicionales que el puesto requiere son:

- Seguir la guía de diseño de casos de prueba.
- Documentar el plan de prueba.
- Automatizar las pruebas que así lo requieran.
- Informar y realizar un seguimiento de los problemas encontrados en el sistema durante el proceso de validación.

Los requisitos solicitados para este puesto se piden:

- Licenciatura en Ingeniería Mecatrónica, Electrónica o en Sistemas.
- Más de 2 años de experiencia relacionada con el puesto.
- Conceptos básicos de los sistemas embebidos.
- Conocimientos de electrónica.
- Conocimientos de programación y automatización (C, Python, LabVIEW u otros lenguajes de programación relevantes).
- Buenas habilidades de escritura, lectura y habla en inglés.

Antecedentes

El proyecto en el cuál trabajé se trata de una unidad de telemática (TCU: *Telematic Control Unit*) para el ámbito automotriz. En la *Figura 3* se muestra un modelo 3D de la unidad de telemática desarrollada por la empresa. La telemática resulta de la unión de los campos de las telecomunicaciones y de la informática; por lo tanto, abarca a la tecnología de envío, recepción y almacenamiento de información a través de dispositivos de telecomunicación. Sin embargo, a pesar de que la telemática es empleada en múltiples ámbitos, es un término que en la actualidad se usa con mayor frecuencia para hacer referencia a las soluciones telemáticas aplicadas en la industria automotriz.



Figura 3: TCU desarrollada por Continental. [8]

La telemática en el ámbito automotriz es un método para recopilar y compartir información que pueden ir desde los datos obtenidos por el GPS, distancia recorrida, hasta incluso el desempeño del vehículo. Esto se logra a través de un sistema electrónico integrado a bordo del vehículo que lo conecta de forma inalámbrica a servicios en la nube o a otros vehículos. [9]

El subsistema para el cual desempeñé mis actividades fue el módulo de WIFI de una unidad telemática. Este módulo es empleado por la TCU para comunicarse a través de redes inalámbricas de banda ancha, ya sea de 2.4GHz o 5GHz. En la *Figura 4* se pueden apreciar esquemáticamente los principales subsistemas de los que se compone la TCU, así como la ubicación del módulo de WIFI en este.

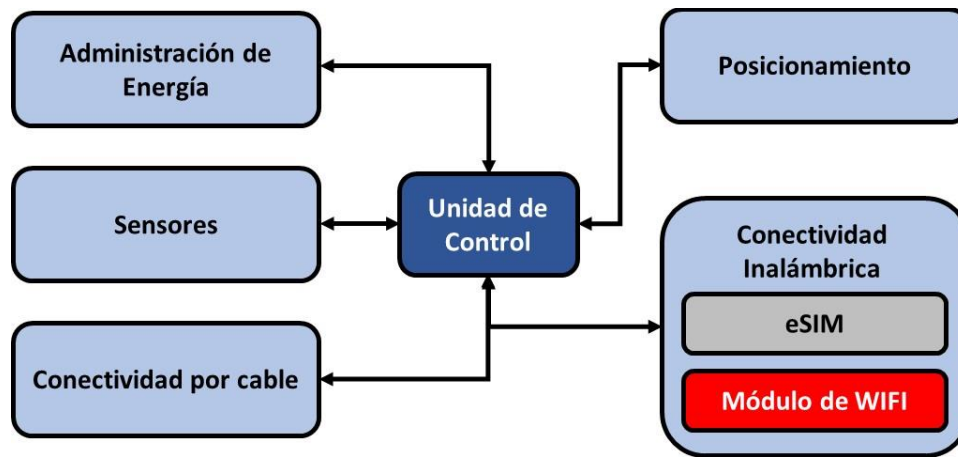


Figura 4: Diagrama general de la TCU. En rojo se resalta el módulo de WIFI. [10]

Para trabajar en este proyecto requerí tener conocimientos primordialmente en Linux, ya que el *software* de la TCU está basado en este sistema operativo; de igual forma, requerí de conocimientos en redes. Por otro lado, para la automatización de las pruebas sobre este dispositivo, necesité de conocimientos en electrónica, en programación, principalmente Python, y en una herramienta interna de la empresa que se utiliza para automatizar el envío de comandos a la unidad telemática a través de comunicación serial.

Algunas otras herramientas que empleé en este proyecto han sido: Android Debug Bridge para establecer la comunicación entre la PC y la unidad telemática; Jenkins para automatizar la ejecución de las pruebas; Selenium para automatizar la configuración del enrutador a través de una página web; fuentes de poder programables con interfaz USB; y PowerShell para hacer depuración manual y ejecutar *scripts* que configuraban automáticamente algunos aspectos de la PC a la que cada TCU estaba conecta.

Definición del problema y contexto de la participación profesional

Cuando me uní al equipo de trabajo, el proyecto ya estaba en desarrollo, por lo que inicialmente participé en sesiones de integración. En estas sesiones, se me explicó el estado actual del proyecto y la forma en que se estaba llevando a cabo el trabajo. Por un lado, estaba el equipo de desarrollo, encargado de preparar periódicamente versiones de *software* para la TCU con las actualizaciones correspondientes. Paralelamente, estaba el equipo de pruebas, al cual me uní, con la responsabilidad de validar el correcto funcionamiento de la TCU utilizando la última versión de *software* proporcionada por el equipo de desarrollo. Además, el cliente, quien había solicitado el proyecto, también realizaba pruebas periódicas utilizando la última versión de *software* validada previamente por nuestro equipo de pruebas. De esta manera, se tenía una doble verificación del correcto funcionamiento y se realizaba una revisión del progreso en el desarrollo. En la *Figura 5* se muestran los diferentes roles de cada integrante y la ubicación de mi puesto dentro del equipo de trabajo:

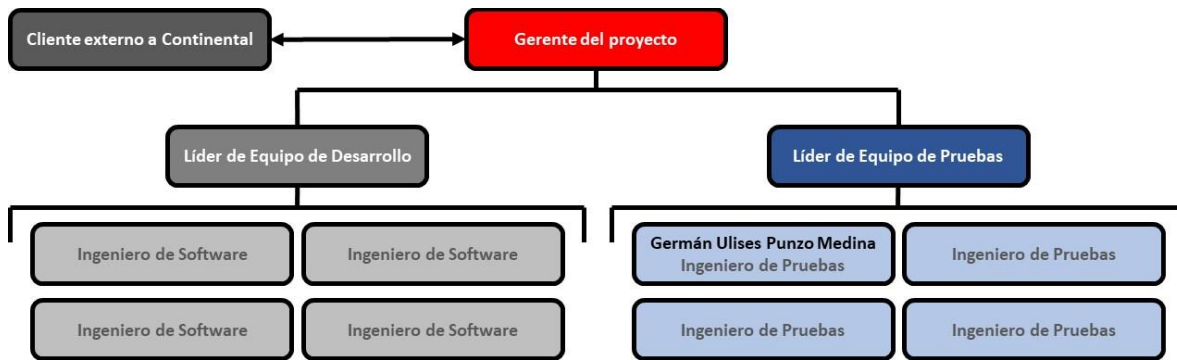


Figura 5: Organigrama del equipo de trabajo. [11]

De igual manera, en dichas sesiones se estableció que mi principal objetivo sería diseñar e implementar las pruebas necesarias para garantizar la calidad del producto con base a los requerimientos relacionados con el módulo de WIFI de la unidad telemática, así como actualizar las pruebas ya existentes conforme a las necesidades del cliente y a los cambios en el *software* de la TCU que el equipo de desarrollo fuera presentando durante el proceso. De esta forma, mi meta quedaría establecida en pasar del 60% al 95% de los requerimientos del cliente cubiertos por al menos un caso de prueba.

Otra tarea que tendría que ser llevada a cabo en paralelo con la anterior, sería automatizar las pruebas y otras actividades pertinentes, de tal manera que el tiempo de involucramiento en la ejecución de las pruebas y recopilación de resultados se reduzca a menos de cinco días hábiles.

Por último, en caso de que se llegase a encontrar alguna falla en el sistema durante la ejecución de las pruebas, ésta tendría que ser reportada detallando resultado esperado, resultado obtenido, condiciones y pasos a seguir para replicar dicha falla. Este reporte se enviaba al equipo de desarrollo con el objetivo de que dicha falla sea corregida antes de que la versión de *software* sea enviada al cliente. Es así como mi tercera meta establecería que el cliente no deberá reportar ninguna falla relacionada con el módulo de WIFI.

Metodología

Metodología de trabajo

Para el desarrollo de nuestro trabajo dentro del equipo de pruebas adoptamos algunas *metodologías ágiles* derivadas de *Scrum*, esto con la finalidad de tener un proceso organizado y buenas prácticas para trabajar colaborativamente, en equipo y obtener mejores resultados. Algunas de las que llevamos a cabo fueron:

- La división de roles: *Product Owner* (Propietario/a del producto) era la gerente del proyecto, quien mantenía comunicación constante con el cliente, y con los líderes de desarrollo y pruebas para darle seguimiento al desarrollo del proyecto; *Scrum Master* fue ejercido por el líder del equipo de pruebas, él era responsable de dar seguimiento al avance de las pruebas, comunicaba los resultados de las pruebas a la *Product Owner* y se encargaba de eliminar los obstáculos que se fueran presentando para el desarrollo de las pruebas; *Developers* éramos los encargados de diseñar y ejecutar las pruebas, y reportar cualquier falla del producto encontrado durante las pruebas.
- *Daily Standup Meetings* (Reuniones diarias de pie): estas reuniones diarias fueron sesiones *online* de máximo media hora en las que cada integrante mencionaba qué hizo el día anterior, qué haría ese día y si tenía algún problema que le impidiera hacer su trabajo.
- *Sprint*: Era el periodo en el cual se llevaban a cabo las pruebas pertinentes a partir de que el equipo de desarrolladores liberaba una nueva versión de *software* para cargar en la TCU. Comprendía dos semanas y al final se entregaba el reporte de pruebas para validar que la nueva versión puede ser entregada al cliente.

La metodología Scrum está enfocada principalmente al desarrollo de *software* [2], sin embargo, en nuestro caso tuvimos que hacer algunas adaptaciones para implementarlo en el ámbito de pruebas.

Plan de pruebas

Un plan de prueba (*Test plan*) documenta el proceso que se utilizará para verificar y garantizar que un producto o sistema cumpla con sus especificaciones de diseño y otros requisitos. Dicho proceso se elige conforme al producto, la responsabilidad de la organización y la fase de desarrollo en la que se encuentre [1]. Dado que el desarrollo de la TCU se encontraba en la etapa de implementación, en la cual el código se había integrado para ejecutarse en el correspondiente *hardware*, el proceso que se llevó a cabo fueron las pruebas de sistema.

Las pruebas de sistema (*System Testing*) son pruebas realizadas en un sistema integrado completo para evaluar el comportamiento del sistema de acuerdo con sus requerimientos

especificados. Su propósito es detectar cualquier defecto en el desempeño tanto entre los subsistemas o módulos integrados como en el sistema como un todo [1]. En mi caso, las pruebas que debía desarrollar, ejecutar y documentar estaban enfocadas en el módulo de WIFI y cómo éste interactuaba con otros subsistemas de la TCU relacionados.

A su vez, el plan de pruebas de WIFI se dividió en tres conjuntos de pruebas (*TS: Test Suite*). Cada *Test Suite* contenía diferentes casos de prueba (*TC: Test Case*) dependiendo del grado de impacto de los cambios implementados en el código por el equipo de desarrollo para la versión de *software* en curso. En la *Figura 6* se sintetiza la división del plan de pruebas.

- En la primera *Test Suite* se agrupaban los *Test Cases* enfocados en validar las principales características y funcionalidades, y se ejecutaba cuando los cambios implementados no afectaban de manera directa dicho subsistema con el objetivo de verificar que las funcionalidades de dicho módulo no se vieron afectadas.
- La siguiente *Test Suite* englobaba todos los casos de prueba para cada uno de los requerimientos del subsistema, por lo que se ejecutaba cuando los cambios implementados afectaban directamente al subsistema en cuestión con el objetivo de verificar que todos los requerimientos se cumplen de manera satisfactoria.
- Y finalmente, la última *Test Suite* que, además de contener los casos de pruebas para los principales requerimientos, agrupaba pruebas específicas para cada cambio que se habían implementado en el correspondiente módulo debido a la corrección de algún defecto previamente encontrado; ésta se ejecutaba posteriormente a que se hubiese reportado algún defecto en el subsistema y corregido por el equipo de desarrollo con el objetivo de verificar que el defecto fue solucionado y no se afectó a otras funcionalidades.

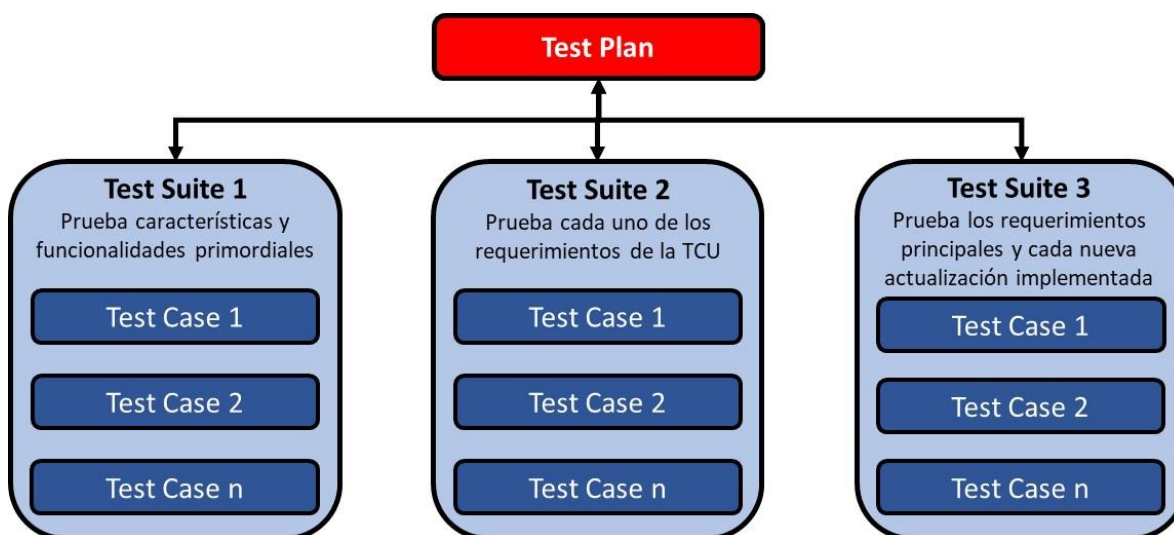


Figura 6: Diagrama del plan de pruebas. [11]

En ingeniería de pruebas, un caso de prueba es un conjunto de pasos, condiciones y resultados bajo los cuales se determina si una característica o funcionalidad de un sistema resulta o no aceptable [1]. Cada una de estas características y funcionalidades que el sistema debe tener es definida en la lista requerimientos desde el inicio del proyecto en común acuerdo entre el cliente y el equipo de desarrollo. Como parte del plan de pruebas del proyecto, se agruparon los requerimientos de acuerdo con el subsistema al que estaban enfocados. A continuación, se enlistan algunas de las características de la TCU relacionadas con el módulo de WIFI que debía tener (Por cuestiones de privacidad, no se dan más detalles sobre el funcionamiento):

1. El módulo de WIFI se puede encender y apagar.
2. El módulo de WIFI se puede establecer como modo cliente.
3. El módulo de WIFI se puede conectar a redes WIFI con encriptación de seguridad.
4. El módulo de WIFI se puede conectar a redes de WIFI abiertas.
5. El módulo de WIFI se puede conectar a redes ocultas y visibles.
6. Para conectarse a una red WIFI, el sistema solicita el SSID (nombre de la red), contraseña, BSSID y tipo de encriptación de seguridad.
7. Una vez conectado a una red WIFI satisfactoriamente, la información de la red (SSID, contraseña, BSSID y tipo de seguridad) es guardada en una lista.
8. Una vez conectado a una red WIFI, se puede establecer comunicación a internet.
9. El módulo de WIFI se puede desconectar de la red a la que esté conectada.
10. Si al tratar de conectarse a una red previamente guardada, se ingresa una contraseña incorrecta, el sistema de WIFI reintentará conectarse corrigiendo el dato con la información guardada para lograr conectarse satisfactoriamente.
11. Si el módulo de WIFI es apagado mientras se está conectado a una red, y posteriormente se enciende de nuevo, el módulo deberá permanecer desconectado de la red.
12. La lista de redes de WIFI guardadas se mantiene aún después de apagar y volver a encender la TCU.

Automatización de las pruebas: Hardware

En los periodos posteriores de haber enviado el reporte de prueba al cliente y antes de iniciar la siguiente ejecución de pruebas con la nueva versión de *software*, mi principal tarea era continuar con el diseño de nuevas pruebas para verificar que la TCU cumple con más elementos de la lista de requerimientos del módulo de WIFI y continuar con la automatización de dichas pruebas y tareas pertinentes.

Para automatizar las pruebas empleé diversas herramientas tanto para el *hardware*, por ejemplo, fuentes de alimentación automatizables, como para el *software* principalmente mediante *scripts*. En la *Figura 7* se muestra un diagrama general de cómo se encontraba conectado el sistema para ejecutar las correspondientes pruebas de WIFI, los elementos utilizados para otras diferentes pruebas se omiten para simplificar el diagrama.

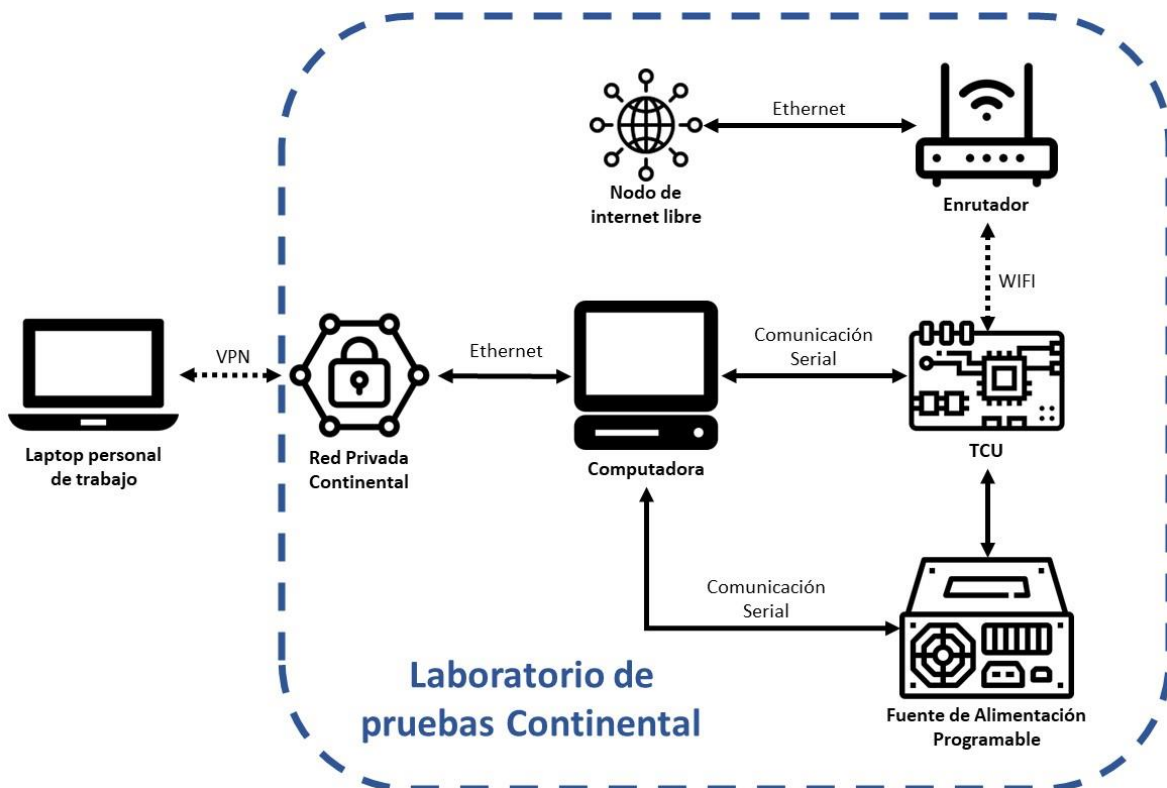


Figura 7: Diagrama general de conexiones enfocado en el equipo de pruebas de WIFI. [12]

La TCU se encontraba conectada con su respectiva computadora mediante comunicación serial a través de un puerto USB. Así mismo, la TCU era alimentada mediante una fuente de poder programable, a la cual se le enviaban comandos por comunicación serial desde la misma computadora para controlar el encendido y apagado de la TCU conforme se necesitara. En el mismo laboratorio, se encontraba el Enrutador conectado a internet al cual las TCU's se conectaban durante las pruebas de WIFI. Por último, la computadora

estaba conectada a una red privada de Continental, de esta forma se podía tener control remoto de esta computadora desde otro equipo de cómputo diferente por medio de un *software* llamado *Remote Desktop*, siempre y cuando estuviera conectado a la misma red, ya sea de forma directa o por medio de una red privada virtual (*VPN: Virtual Private Network*).

Automatización de las pruebas: Software

Para la edición y ejecución de los casos de prueba, se contaba con un *software* de automatización de pruebas interno (*ATS: Automation Test Software*). Con este *software* se abrían archivos XML relacionados con cada *Test Suite* y en su interfaz gráfica se desplegaban los TC's contenidos. Una vez abierta una *Test Suite*, se podían crear, editar o eliminar sus *Test Cases*. Para editar cada paso de un TC, se contaba con diferentes opciones tales como enviar mensajes por comunicación serial a la TCU, esperar una respuesta de la TCU, manejo de variables, condicionales, ejecución de comandos en *CMD*, entre algunas otras. Estas acciones eran ordenadas secuencialmente y se agrupaban dentro de cada TC. Así mismo, el ATS podía ejecutar los TC's de la TS y, posteriormente, generar un reporte donde se detallaba si algún TC había fallado debido a un comportamiento no esperado de la TCU.

Por otro lado, contaba con un *script* de Python el cual se conocía como Ejecutor de Pruebas de WIFI (*Ejecutor* de ahora en adelante) debido a la cantidad e importancia de las tareas que éste realizaba. Para comenzar, recibía tres argumentos: el primero era la versión de *software* con la que debía probarse la TCU; el segundo era el número de la *Test Suite* que sería ejecutada; y el tercero definía qué configuración WIFI del enrutador sería probada, es decir, si serían redes visibles u ocultas y qué tipo de encriptación de seguridad.

La primera acción del ejecutor era encender la TCU mandando un comando a la fuente de poder. Después, mediante comunicación serial, revisaba la versión de *software* que estaba cargada en la TCU; en caso de que no fuera la solicitada, procedía a *flashear* la versión correcta, la cual era descargada desde una *unidad compartida* donde el equipo de desarrollo la había guardado. A continuación, el Ejecutor corría un *script* de Selenium que establecía vía web la configuración WIFI del enrutador. Posteriormente, el ejecutor abría una instancia del ATS y se ejecutaba la TS seleccionada. Al terminar la ejecución de las pruebas, el reporte generado se subía automáticamente a la unidad compartida. En caso de que se hubiera solicitado probar más de una configuración WIFI, el Ejecutor establecía la siguiente configuración del enrutador y ejecutaba nuevamente las pruebas. Finalmente, el Ejecutor apagaba la TCU y cerraba toda instancia y puerto utilizado. En el diagrama de flujo de la *Figura 8* se describe de forma general el proceso de este *script*.

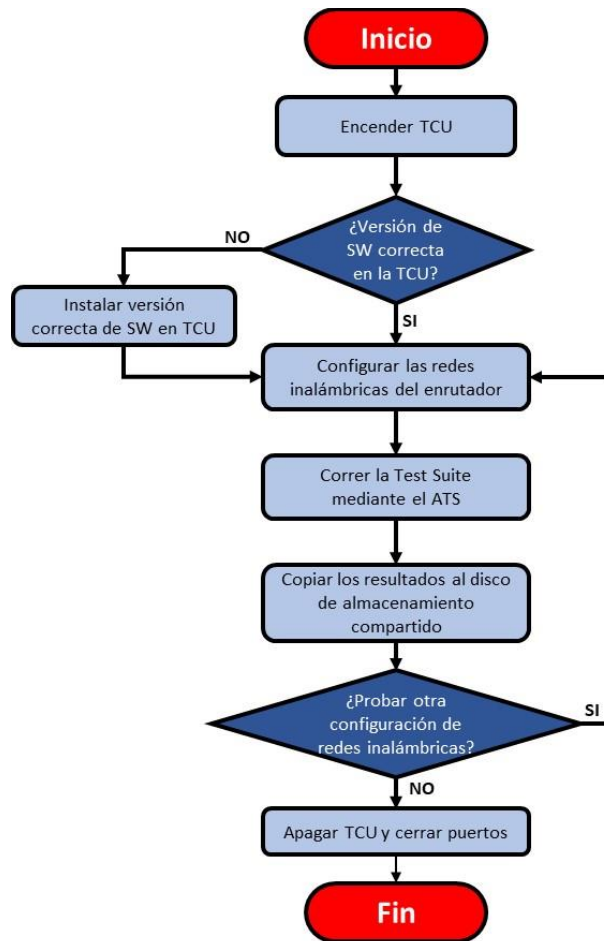


Figura 8: Diagrama de flujo del Script Ejecutor de pruebas de WIFI. [12]

Para ejecutar las pruebas por medio de este *script*, era necesario hacerlo en la computadora del laboratorio conectada a cada TCU, ya sea en persona o remotamente mediante *Remote Desktop*. En total, se llegó a contar en el laboratorio con hasta cinco TCU's conectadas con sus respectivas computadoras para correr las pruebas para todos los sistemas. A pesar de eso, eran insuficientes para correr las pruebas necesarias durante el horario de trabajo, ya que además de las pruebas de WIFI, mis compañeros de equipo debían probar otros subsistemas de las mismas TCU's y los tiempos para cada *Test Suite* podían ser mayores a 5 horas. Para solucionar esto, se empleó Jenkins, un servidor de automatización.

Jenkins era utilizado para hacer una cola de todas las TS que serían ejecutadas, y se conectaba a todas las computadoras del laboratorio para distribuir automáticamente dichas ejecuciones. El proceso de ejecución de Jenkins fue programado mediante un *script* en Groovy, el cual recibía como argumentos qué TS ejecutaría y sus respectivos parámetros establecidos por quien ejecutó la prueba desde la interfaz de Jenkins. Lo primero que hacía el *script* de Groovy era seleccionar una computadora del laboratorio que estuviera disponible para ejecutar las pruebas. Una vez que ingresaba a la

computadora, actualizaba mediante GIT todos los cambios realizados a las pruebas y los respectivos recursos dentro del repositorio. Posteriormente, corría el Ejecutor del sistema seleccionado y enviaba los parámetros establecidos desde Jenkins. Mientras una prueba se encontraba corriendo, Jenkins revisaba si hay otra prueba en espera y, siempre y cuando otra computadora estuviera disponible, se ejecutaba en ella la siguiente *Test Suite*. Al terminar la ejecución en una computadora, ésta quedaba disponible. De esta forma, las pruebas se ejecutaban incluso durante la noche, y al día siguiente sólo se tenía que revisar los resultados obtenidos y hacer el reporte correspondiente.

Mis actividades

Capacitación

Mis primeras actividades se centraron en tomar capacitaciones durante dos semanas. De todas estas capacitaciones, las más importantes para llevar a cabo mis tareas en el proyecto fueron los cursos sobre Linux para Sistemas Embebidos, Git, Jenkins y Selenium con Python. Posteriormente, asistí a sesiones en las que me explicaron en detalle las funcionalidades de la TCU, cómo establecer comunicación con ella, cómo cargar manualmente cualquier versión de *software* y cómo controlar sus funciones de WIFI. Finalmente, me enseñaron a utilizar herramientas de automatización de pruebas, ejecutar las pruebas y crear nuevas pruebas mediante el ATS.

Conexión de equipo de prueba

En un principio se contaba con dos TCU conectadas al equipo de prueba; sin embargo, a medida que avanzaba el tiempo, se incrementó la cantidad y el nivel de pruebas necesarias para el sistema completo. Debido a esto, se tuvieron que añadir tres TCU más con sus respectivos equipos. Yo participé en la conexión de uno de los sistemas, además de configurar su respectiva computadora instalando todos los programas y controladores necesarios. En la sección de Automatización de Pruebas, se describe detalladamente el equipo utilizado y sus conexiones para automatizar el proceso de prueba en cada TCU.

Actualización de scripts de automatización

Por otro lado, en el laboratorio de pruebas contábamos con un enrutador inalámbrico que era capaz de transmitir simultáneamente en dos bandas de frecuencia: 2.4GHz y 5GHz. Además, se podía configurar para que estas redes fueran visibles u ocultas. También era posible establecer una contraseña para cada una de ellas, ya sea mediante seguridad WPA2 o como redes abiertas. La gran ventaja de utilizar este enrutador era que se podían definir las características de las redes inalámbricas a través de un sitio web. Esto significaba que se podía configurar desde cualquier computadora con acceso a Internet, incluso fuera del laboratorio, siempre y cuando se contara con las credenciales de acceso a la cuenta vinculada al enrutador. Gracias a esta funcionalidad, era posible establecer automáticamente la configuración deseada mediante un *script* de Selenium. Dicho *script*

se encargaba de abrir la página web en el navegador de forma automática, ingresar las credenciales y luego establecer la configuración según los parámetros recibidos.

Sin embargo, en un principio este *script* solía fallar en ocasiones, lo que resultaba en una configuración incorrecta del enrutador. Por lo tanto, realicé ajustes en el *script* para mejorar el método de búsqueda de los elementos en la página web. Además, implementé el manejo de errores y reintentos en caso de que ocurriera un *timeout*. Esto último podía ocurrir si la velocidad de internet era demasiado lenta, lo cual provocaba que fallara la búsqueda automática de los elementos de la página web. Además, cada vez que se realizaba una actualización en el sitio web, mi tarea consistía en realizar los ajustes necesarios en el *script* para mantenerlo en funcionamiento.

Los datos para la configuración del enrutador se encontraban en una lista dentro del *script* ejecutor. A continuación, se enlistan las diferentes configuraciones de las redes inalámbricas del enrutador con las que se debían ejecutar las pruebas de WIFI:

1. Redes visibles y seguridad WPA2.
2. Redes ocultas y seguridad WPA2.
3. Redes visibles y abiertas.
4. Redes ocultas y abiertas.

Estas cuatro configuraciones se probaban cada vez que el *script* Ejecutor corría. Sin embargo, en ocasiones sólo era necesario probar una de estas configuraciones. Por lo tanto, realicé ajustes para que el Ejecutor pudiera recibir como argumentos qué número configuración se requería probar o si se ejecutasen las cuatro configuraciones. Además, agregué la posibilidad de recibir un valor de -1 como argumento, lo que permitía ejecutar una configuración aleatoria. Esta funcionalidad resultaba muy útil para pruebas de validaciones rápidas.

El *script* de configuración del enrutador se encargaba de acceder al sitio web y realizar los cambios necesarios en los campos específicos para cada prueba. Sin embargo, había otros campos que debían mantenerse constantes en cada prueba y no debían ser modificados por el *script*, como el nombre de las redes, el canal de transmisión y el estándar de red utilizado, entre otros. Mas esto presentaba un inconveniente, ya que, si esos campos se modificaban manualmente o se restablecían a los valores de fábrica, posteriormente debían ser configurados manualmente nuevamente según los requisitos de las pruebas. Por lo tanto, actualicé el *script* para que verificara y configurara automáticamente esos campos en caso de ser necesario. De esta manera, si se producían cambios no deseados en esos campos, el *script* podía restablecerlos a los valores requeridos para las pruebas.

En un principio, las pruebas de WIFI tenían que ser ejecutadas directamente en una de las computadoras del laboratorio utilizando el *script* Ejecutor, y no desde Jenkins como se hacía con las pruebas de otros subsistemas de la TCU. Por lo tanto, mi tarea consistió en actualizar el código necesario en el *script* de Groovy para que las pruebas de WIFI

corrieran automáticamente desde Jenkins. De igual forma, tuve que crear un *Job*, es decir un espacio en el servidor de Jenkins que contiene campos y que, al ejecutarse, éstos se envían como argumentos al *script* de Groovy vinculado. Algunos de los campos para las pruebas de WIFI son la versión de *software*, la configuración del enrutador requerida y el nivel de prueba. Después de realizar estas actualizaciones, las pruebas de WIFI pudieron ser ejecutadas desde Jenkins y se pudieron aprovechar sus beneficios, como la ejecución automática de las pruebas en las computadoras del laboratorio o que, si en el momento no hubiera alguna computadora disponible, la ejecución quedara a la espera hasta que alguna computadora se liberara y comenzara automáticamente la ejecución.

Diseño e implementación de pruebas

Cuando empecé a trabajar en este proyecto, solo se habían cubierto el 60% de los requerimientos de WIFI. Esto significa que las pruebas existentes solo verificaban el correcto funcionamiento del 60% de los comportamientos solicitados por el cliente para este subsistema. Por lo tanto, mi tarea principal fue cubrir al menos el 95% de los requerimientos restantes. Para lograrlo, tuve que crear al menos un caso de prueba para cada uno de los requerimientos que aún no se habían cubierto. Cada caso de prueba debía incluir una descripción paso a paso de la prueba realizada para verificar la correcta funcionalidad de una característica específica de la TCU, basándose en la definición de los requerimientos. Es posible que se necesitara más de un caso de prueba para verificar un solo requerimiento, dependiendo de la complejidad del requerimiento en cuestión.

A continuación, para ejemplificar, se describe el caso de prueba que verifica si la TCU se enciende correctamente. Es importante destacar que cada caso de prueba se ejecuta bajo condiciones previas; en este caso, se asume que la TCU está previamente conectada a todos los componentes de prueba, como a la fuente de alimentación y a la computadora que muestra la información que la TCU genera. Cabe aclarar que la siguiente descripción no incluye detalles específicos por razones de privacidad.

1. Encender la fuente de alimentación que suministra energía a la TCU con el voltaje correspondiente. Esto se logra enviando el comando adecuado a través de la comunicación serial con la fuente de alimentación.
2. Esperar a que la TCU comience a desplegar datos de encendido. Esto se verifica a través de la comparación del texto impreso y el texto esperado, ya sea mediante valores constantes como el nombre de la empresa o mediante *expresiones regulares* como la fecha que puede cambiar, pero mantiene un formato constante.
3. Corroborar que el tiempo de arranque es menor a 60 segundos. Esto significa que el tiempo que corre desde que se enciende la fuente de poder hasta que la TCU termina de desplegar los datos de encendido es menor a un minuto.
4. Verificar que la versión de *software* desplegada sea correcta. Es decir, dentro de los datos impresos por la TCU al encender, se debe mostrar la versión de *software* que corresponde a la versión que se está probando.

5. Enviar un comando a través de la comunicación serial a la TCU y confirmar que responda correctamente. Es decir, se envía un comando específico que debería hacer que la TCU despliegue un mensaje determinado, lo que confirma que está funcionando correctamente.

De manera similar, los demás casos de prueba son descritos y posteriormente ejecutados mediante el ATS. Como se mencionó anteriormente, esta herramienta cuenta con las funcionalidades necesarias para llevar a cabo las pruebas tales como el envío de comandos por comunicación serial, buscar cadenas de texto o expresiones regulares en mensajes recibidos por comunicación serial, entre algunas otras.

Teniendo en cuenta lo anterior y analizando cada uno de los requerimientos pendientes de verificación, procedí a crear los casos de prueba pertinentes. Esta tarea fue la que requirió más durante mi participación, ya que implicaba un buen conocimiento del producto y la ejecución constante de las pruebas para asegurarse de que estuvieran diseñadas de manera correcta y fueran suficientes, evitando así *falsos positivos* o *falsos negativos*. En caso de tener dudas acerca de la descripción de los requerimientos, acudía ocasionalmente al equipo encargado de definir y redactar dichos requerimientos para obtener aclaraciones. Además, mantuve una comunicación constante con el equipo de desarrolladores para resolver dudas durante la implementación de las pruebas, especialmente para entender con mayor detalle el comportamiento esperado de la TCU.

Para el diseño de los casos de prueba, se analizó el sistema en su conjunto como una caja negra, es decir, no nos involucramos con el código ni los detalles internos del *hardware*, sino que nos enfocamos únicamente en las entradas y salidas esperadas [1]. Para llevar a cabo las pruebas, utilicé las herramientas disponibles en el ATS, y cada caso de prueba fue integrado en la *Test Suite* de nivel medio, la cual debía incluir un caso de prueba para cada requisito de WIFI. Después de varios meses de trabajo, logré cubrir el 97% de los requisitos. El 3% restante fue descartado para ser probado debido a diversas circunstancias que estaban fuera de nuestro alcance, como posibles modificaciones en el *hardware*.

De todos los requisitos de WIFI, hubo un par que no se pudieron cubrir únicamente con las herramientas del ATS. Uno de ellos era el *Throughput performance test*, que consistía en medir el rendimiento de la transmisión de paquetes de datos enviados por la TCU a través de WIFI [13]. La TCU contaba con la herramienta *Iperf3*, la cual se utiliza para realizar pruebas en redes informáticas y puede funcionar como cliente o servidor, permitiendo medir el rendimiento entre ambos extremos de la comunicación. En un principio, esta prueba se ejecutaba manualmente utilizando dos TCU's conectadas a un mismo enrutador, como se muestra en la *Figura 9* lado izquierdo. Una TCU actuaba como servidor y la otra como cliente, y se realizaba la prueba. Sin embargo, esta configuración resultaba en velocidades de transmisión muy bajas.

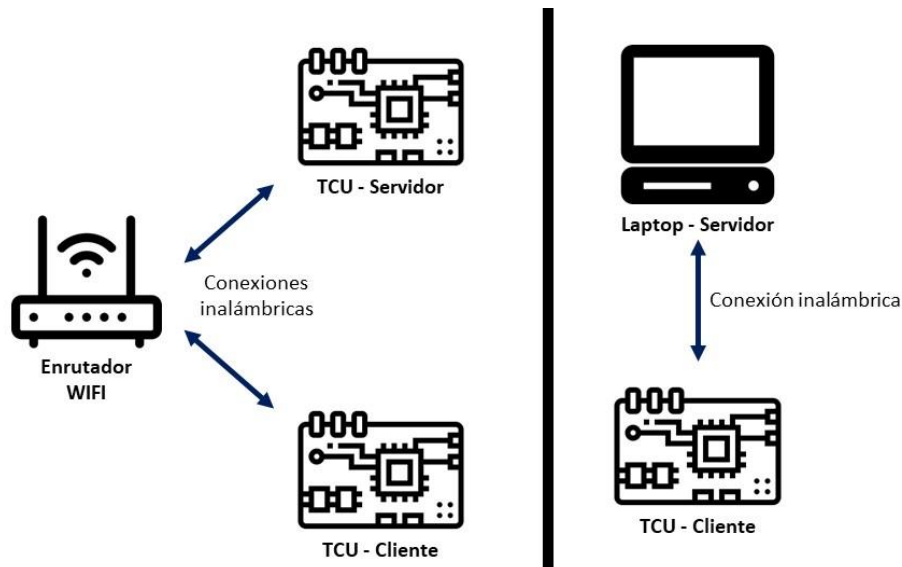


Figura 9: Conexión anterior para prueba de Throughput a la izquierda (TCU-Enrutador-TCU) y nueva conexión a la derecha (TCU-Laptop). [12]

Por lo anterior, mi tarea consistió en encontrar una nueva forma de ejecutar esta prueba con el fin de aumentar la velocidad de transmisión y, sobre todo, asegurarme de que pudiera ejecutarse de manera automática dentro de las *Test Suites*. Dadas las características del problema, se me ocurrió utilizar el *hotspot* de las laptops a las que estaban conectadas las TCU, de esta manera se podría usar la TCU como cliente y la laptop como servidor. Por lo tanto, tuve que instalar el programa Iperf3 también en dichas laptops. De este modo, la transmisión se realizaba directamente de dispositivo a dispositivo, sin pasar por un punto intermedio como ocurría en la configuración anterior. En la *Figura 9* lado derecho se muestra cómo se conectaría la TCU al *hotspot* de su laptop. Gracias a esta nueva configuración, la velocidad de transmisión se incrementó cuatro veces, alcanzando así los valores esperados por el cliente.

Mi siguiente tarea fue encontrar una forma de automatizar la prueba anterior. Investigué cómo podía encender y configurar el *hotspot* de la laptop a través de un *script* y descubrí que era posible automatizar algunas funcionalidades de Windows utilizando Powershell. Así mismo, encontré los comandos necesarios para programar el mencionado *script* y poder configurar y encender el *hotspot*. Además, programé otro *script* que corría automáticamente Iperf3 en la computadora, manteniéndola en modo servidor hasta que terminara la prueba. Una vez finalizados estos *scripts*, los añadí al repositorio de archivos de las pruebas para que pudieran ser ejecutados utilizando las funciones del ATS dentro de las *Test Suites*.

El otro requerimiento que necesitaba herramientas adicionales era el que especificaba la capacidad de la TCU para conectarse a un enrutador con seguridad WEP. El problema era que no teníamos un enrutador con estas características en el laboratorio, por lo que tuve que buscar uno que pudiéramos adquirir y utilizar para cubrir este requerimiento. Una vez

recibí el enrutador mencionado, lo conecté en el laboratorio y verifiqué su funcionamiento realizando pruebas manuales con la TCU. También tuve que actualizar el *script* Ejecutor de WIFI para agregar una nueva configuración a probar. Es importante destacar que la configuración de este enrutador no podía ser modificada a través de una página web, por lo que se mantuvo constante como: Redes visibles (2.4GHz y 5GHz) y seguridad WEP.

La tarea que realizaba con mayor frecuencia era ejecutar las pruebas de validación. Regularmente, el equipo de desarrollo generaba nuevas versiones de *software* que debían ser probadas antes de ser entregadas al cliente. Dependiendo de los cambios realizados entre cada versión, se determinaba el nivel de pruebas que debía ser ejecutado. Incluso cuando los cambios no estaban directamente relacionados con las funciones de WIFI, era necesario realizar una prueba de regresión con todas las configuraciones de enrutador para verificar que los cambios no afectaran de manera indirecta estas funciones. Además, cuando se realizaban modificaciones directas en las funciones de WIFI, era necesario actualizar las Suites de Pruebas, ya sea para modificar algún caso de prueba existente o para agregar nuevos casos si se trataba de una nueva función. Una vez que las *Suites* de Pruebas estaban actualizadas, procedía a ejecutar las pruebas para validar la nueva versión de *software*.

Reporte de resultados de pruebas

Con el apoyo de Jira, un *software* de gestión de proyectos, se dio seguimiento a la ejecución de las pruebas, documentábamos los resultados y reportábamos cualquier error que se encontrara. En caso de identificarse algún error, se debía crear un *ticket* en Jira, describiendo los pasos necesarios para reproducir el error, la versión de *software* utilizada, el resultado obtenido y el resultado esperado. Después de generar este *ticket* con la información correspondiente, se asignaba al equipo de desarrollo para que trabajaran en la solución. Una vez que integraban la solución, mi tarea era ejecutar una prueba para confirmar que el error ya no estuviera presente. Una vez que todas las pruebas se habían ejecutado, procedía a reportar los resultados finales con ayuda de un *software* interno que generaba el informe final de prueba que sería enviado al cliente.

Otras actividades

Además de las actividades mencionadas, también participé en algunas tareas adicionales. Una de ellas fue la creación de un contenedor de *software* utilizando Docker en un servidor virtual, con el fin de configurar un servidor de GitLab para el equipo de pruebas. También formé parte del equipo de entrevistas para los candidatos que solicitaban una vacante en el equipo de pruebas. Durante las entrevistas, mi tarea principal consistía en evaluar las habilidades de programación de los candidatos.

Resultados

A continuación, se resumen los resultados obtenidos durante mi participación en el proyecto:

1. Diseñar implementar y ejecutar los casos de prueba necesarios para pasar del 65% al 97% de cobertura de los requerimientos de WIFI.
2. Realizar las conexiones necesarias del equipo de prueba en el laboratorio para llevar a cabo la automatización de las pruebas de WIFI.
3. Crear y editar *scripts* mediante Python, Selenium, Groovy y PowerShell para automatizar algunas tareas y la ejecución de pruebas de WIFI.
4. Reducir el tiempo de ejecución de pruebas en un 75% a partir de automatizar tareas que anteriormente eran ejecutadas de forma manual y de optimizar pruebas.
5. Reportar resultados de pruebas de WIFI y dar seguimiento a errores encontrados.

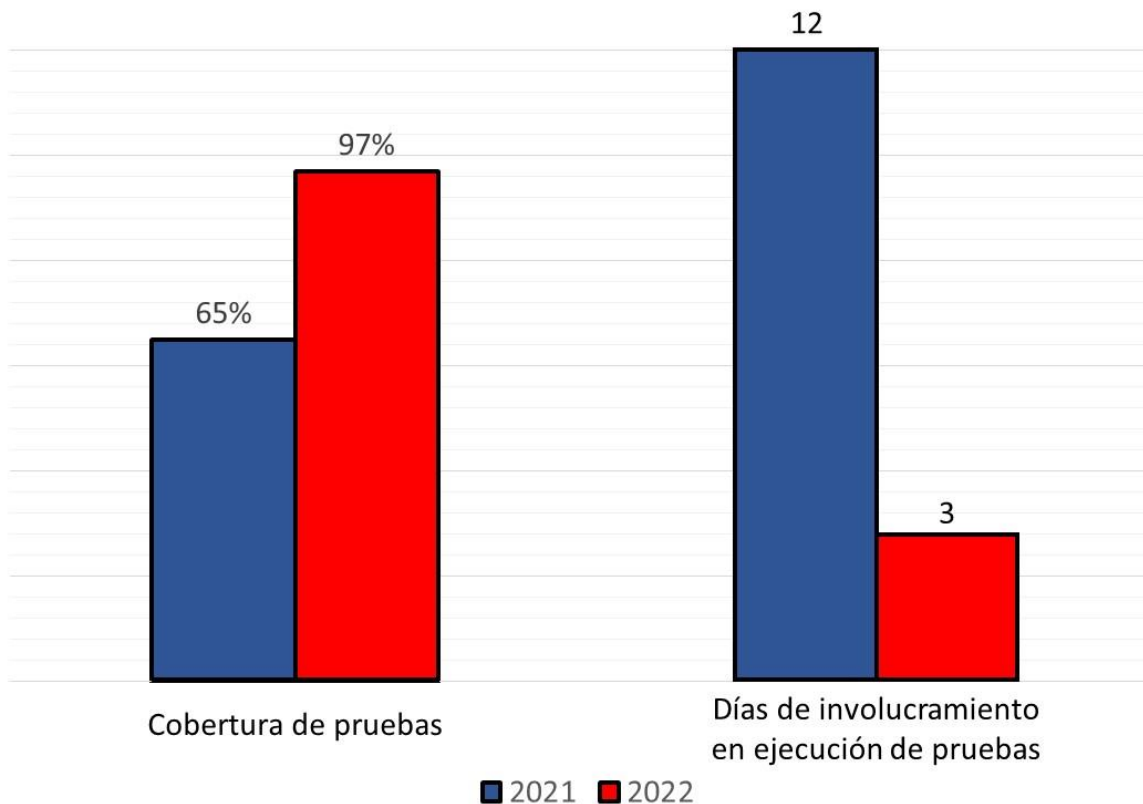


Figura 10: Comparación de los dos principales datos (Cobertura y tiempo de ejecución de pruebas) al inicio y al final de la participación en el proyecto. [12]

Conocimientos de la carrera aplicados en el campo laboral de un Ingeniero de Pruebas

Sin duda los conocimientos adquiridos durante la carrera me dieron la base para poder desempeñar mis actividades de forma efectiva, sobre todo porque en los últimos semestres de la carrera cursé materias obligatorias y optativas enfocadas a la programación y automatización. Esto no sólo contribuyó a adquirir los conocimientos técnicos, sino que me ayudó a desarrollar un pensamiento analítico para encontrar la forma en que una o varias tareas podrían ser automatizadas.

Algunas de las asignaturas que principalmente me proporcionaron conocimientos relevantes para el puesto de ingeniero de pruebas fueron, en primer lugar, aquellas relacionadas con la programación, como 'Computación para Ingenieros', 'Técnicas de Programación' y 'Temas Selectos de Programación'. En estas materias aprendí una serie de conceptos y técnicas que me sirvieron como base para adentrarme en esta área y, posteriormente, aplicar esos conocimientos en mi puesto de trabajo. Por ejemplo, pude crear *scripts* y otros programas para automatizar algunas tareas, así como también entender de manera más sencilla el funcionamiento del *software* de la Unidad Telemática.

En segundo lugar, las materias relacionadas con electrónica, como 'Electrónica Básica', 'Instrumentación' y 'Circuitos Digitales', me proporcionaron los conocimientos necesarios para comprender el funcionamiento de las Unidades Telemáticas. Además, me permitieron diseñar los circuitos pertinentes para probar estos dispositivos, trabajar con ellos y utilizar instrumentos de laboratorio relacionados con esta área.

Finalmente, ciertas asignaturas como 'Diseño Mecatrónico', 'Robótica', 'Control Automático' y 'Control Avanzado' me permitieron profundizar en mi comprensión y aplicación de mis conocimientos en programación y electrónica a través del desarrollo de proyectos que integraban de manera conjunta dichas áreas.

Por otro lado, me gustaría destacar que las actividades que llevé a cabo durante mi servicio social en séptimo semestre en un proyecto de domótica en el Centro de Investigación Avanzada en la Facultad de Ingeniería me fueron de mucho provecho, ya que en ese momento me ayudaron a decidir por el área a la cual me quería especializar, y además adquirí conocimientos teóricos y prácticos sobre electrónica y programación, los cuales me fueron muy útiles durante mis primeras experiencias laborales.

La formación adquirida a lo largo de la carrera y el servicio social fue fundamental para realizar efectivamente mis actividades en el trabajo. Sin embargo, cabe mencionar que hubo otras habilidades que también me fueron de gran ayuda durante mi experiencia laboral, y que las adquirí en actividades extracurriculares. Una de esas actividades fue mi participación en la agrupación Velomóvil RS donde desarrollamos un proyecto que fue presentado internacionalmente. En dicho proyecto trabajé durante un año con un grupo

de personas de diferentes carreras, en donde cada integrante tenía diferentes tareas, pero compartíamos una meta en común. En ese proceso llevé a cabo algunos procesos que posteriormente empleé durante mi experiencia laboral, tales como los planteamientos iniciales del problema, la división del trabajo, encontrar las fortalezas y oportunidades del equipo, reuniones periódicas para presentar avances y problemas del proyecto, gestión de recursos, entre algunos otros.

Por último, me gustaría resaltar lo importante que fue aprender inglés en la ahora Escuela Nacional de Lenguas, Lingüística y Traducción (antes Centro de Enseñanza de Lenguas Extranjeras), ya que es una habilidad que de entrada las empresas toman en cuenta durante las entrevistas de reclutamiento, pero además es sumamente importante durante el trabajo. En mi caso, me fue indispensable para llevar a cabo mis actividades en el proyecto, ya que las personas con las que tuve que interactuar provenían de diferentes países como Alemania, Estados Unidos, Rumania y Japón, y la lengua común para comunicarnos fue el inglés, además de que algunas capacitaciones que tomé y documentación que utilicé se encontraban en este idioma.

Conclusión

Cuando recién me gradué, el puesto de Ingeniero de Pruebas no me era conocido. Sin embargo, cuando comencé a buscar oportunidades laborales, me percaté de que cumplía con los requisitos solicitados y, dado que actualmente existe una considerable demanda en este campo, los beneficios ofrecidos resultaban atractivos. Aunque es importante destacar que la mayoría de las oportunidades se encuentran fuera de la Ciudad de México.

Personalmente, considero que la actividad más interesante en este puesto fue la automatización de tareas, ya que me permitió aplicar numerosos conocimientos adquiridos durante la carrera. Estos incluyen la programación de *scripts*, la instrumentación, así como el diseño e implementación de circuitos. Por otro lado, debo mencionar que la tarea que encontré más desafiante fue la documentación, ya que carecía de suficiente experiencia en este aspecto. Para mejorar en este sentido, me enfoqué en dos áreas principales: la estructuración del contenido y la redacción, tanto en español como en inglés.

Si bien, cuando ingresé al puesto mencionado, pude desarrollar mi trabajo de forma satisfactoria, existen algunas habilidades importantes que, en mi experiencia, podrían complementar el perfil del ingeniero egresado para el campo laboral. Por un lado, se encuentran habilidades técnicas como el uso de *software* de control de versiones (Git), profundizar en el manejo de equipos de laboratorio y el uso de *software* para la gestión de proyectos. Por otro lado, están las habilidades blandas tales como el conocimiento de metodologías ágiles, liderazgo, la elaboración de documentación y la formulación de requerimientos de proyectos.

A partir de mi experiencia en el campo laboral, puedo constatar que el perfil de egresado de la carrera de Ingeniería Mecatrónica de la Facultad de Ingeniería es adecuado para el puesto de Ingeniero de Pruebas, especialmente en empresas que desarrollan proyectos multidisciplinarios que involucran áreas como la computación, la electrónica o la automatización. Sin embargo, cabe señalar que es importante tomar asignaturas relacionadas con este sector en los módulos optativos para complementar de mejor manera el perfil, así como participar en actividades adicionales disponibles en la facultad, como agrupaciones estudiantiles o cursos intersemestrales que complementen el perfil del egresado.



Figura 11: Compañeros ingenieros de prueba en Continental Guadalajara.

Referencias

- [1] Cerquozzi, R. & Decoutere, W. (2023). *International Software Testing Qualifications Board: Foundation Level Syllabus v4.0*. Disponible en:
<https://www.istqb.org/certifications/certified-tester-foundation-level>
- [2] Schwaber, K. & Sutherland, J. (2020) *La Guía Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*. Disponible en:
<https://scrumguides.org/scrum-guide.html>
- [3] Continental. (Enero, 2024). *Continental: Estudios y publicaciones*. Disponible en:
<https://www.continental.com/de/presse/studien-publikationen/>
- [4] Continental Engineering Services. (Diciembre, 2023). *CES: Historia*. Disponible en:
<https://conti-engineering.com/history/>
- [5] Continental. (Diciembre, 2023). *Continental México*. Disponible en:
<https://www.continental.com/es-mx/compania/continental-mexico/>
- [6] Brejnev. (2010). *Continental AG global locations*. Disponible en:
https://en.m.wikipedia.org/wiki/File:Continental_AG_global_locations.png
- [7] Continental Engineering Services. (Diciembre, 2022). *Organigrama de Continental Engineering Services* [Datos no publicados]. Archivo de Continental Engineering Services.
- [8] Continental Engineering Services. (Marzo, 2024). *Telematic Unit*. Disponible en:
<https://conti-engineering.com/components/telematic-unit/>
- [9] Infineon. (Abril, 2023). *Automotive telematics control unit (TCU)*. Disponible en:
<https://www.infineon.com/cms/en/applications/automotive/automotive-security/telematics-control-unit/>
- [10] Continental Engineering Services. (Diciembre, 2022). *Ficha de datos de TCU* [Datos no publicados]. Archivo de Continental Engineering Services.
- [11] Continental Engineering Services. (Diciembre, 2022). *Plan de pruebas de TCU* [Datos no publicados]. Archivo de Continental Engineering Services.
- [12] Continental Engineering Services. (Diciembre, 2022). *Documentación de pruebas de TCU* [Datos no publicados]. Archivo de Continental Engineering Services.
- [13] Netally. (Mayo, 2023). *Network Performance*. Disponible en:
<https://www.netally.com/network-performance/>