



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

ACKLEN AVENUE | HERO UNIT

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11541234 JOSÉ CARLOS ROJAS HERRERA

ASESOR: LIC. TANIA LUCILA MEZA AMADOR

CAMPUS TEGUCIGALPA; ENERO, 2021

RESUMEN EJECUTIVO

El presente informe se entrega como requisito para optar al título profesional de Ingeniero en Sistemas Computacionales, donde se describen las actividades desarrolladas durante la práctica profesional realizada de julio del 2020 a diciembre del 2020. La práctica se realizó en el área de control de calidad de la empresa Acklen Avenue | Hero Unit, en el puesto de Ingeniero en Control de Calidad, cumpliendo con tareas de apoyo a los equipos de desarrollo tales como identificación, documentación y seguimiento de errores, ejecución de pruebas manuales y colaboración constante en el proceso de desarrollo ágil de software.

El control de calidad es un elemento vital en todo proyecto de desarrollo de software si se busca entregar un producto con un estándar de calidad elevado. Existen técnicas y procedimientos comúnmente adoptados por la industria y que en Acklen Avenue | Hero Unit no son la excepción. Estos varían según la naturaleza de los proyectos y las necesidades de los clientes, lo que también lleva a utilizar diferentes recursos y herramientas según sean necesarias.

A lo largo del informe se describen las herramientas que fueron utilizadas, los procedimientos que se siguieron y la metodología que fue implementada en los 4 proyectos en los que se realizó trabajo durante la práctica: Matrix, Glee2, Recruitment y Acera. Asimismo, se describe el aprendizaje conseguido a través del periodo de entrenamiento e introducción a la empresa y el desarrollo profesional ofrecido por la misma para fortalecer los conocimientos del puesto empleado. Finalmente, se mencionan algunas actividades de carácter organizacional que ayudaron a adquirir experiencia en cuanto a forjar relaciones con los clientes y ser parte del engranaje de una empresa de desarrollo de software.

ÍNDICE DE CONTENIDO

I.	Introducción.....	1
II.	Generalidades de la Empresa.....	3
2.1	Descripción de la Empresa	3
2.2	Descripción del Departamento	4
2.3	Objetivos del Puesto.....	5
2.3.1	Objetivo General.....	5
2.3.2	Objetivos Específicos	5
III.	Marco Teórico.....	6
IV.	Desarrollo	21
4.1	Descripción del Trabajo Realizado	21
4.1.1	Entrenamiento: Pathway Orientation	21
4.1.2	Entrenamiento: Boarding Pass Pathway	22
4.1.3	Desarrollo Profesional: Associative Quality Engineer I Pathway	31
4.1.4	Proyectos: Matrix, OIW y Acera	33
4.1.5	Acklen Avenue y Hero Unit	43
4.1.6	Puesto como consultor en PICA.....	44
4.2	Cronograma de Actividades	45
V.	Conclusiones	47
VI.	Recomendaciones	48

Bibliografía..... 49

ÍNDICE DE ILUSTRACIONES

Ilustración 1 – Ejemplo de Muro Kanban	9
Ilustración 2 – Manifiesto Ágil	12
Ilustración 3 – Los 12 Principios del Desarrollo Ágil de Software	13
Ilustración 4 – Evidencia de video instructivo en Pathway Orientation	21
Ilustración 5 – Evidencia de video elaborado para clase “Welcome to Our Family”	23
Ilustración 6 – Temas desarrollados en la clase “Building Community”	24
Ilustración 7 – Temas desarrollados en la clase “Meet the Team”	25
Ilustración 8 – Temas desarrollados en la clase “Culture Champion”	26
Ilustración 9 – Temas desarrollados en la clase “Our Vision and Values”	27
Ilustración 10 – Tarea para la clase “Introduction to Client Expectations”	28
Ilustración 11 – Plan de presencia elaborado para la clase “Remote Professional”	29
Ilustración 12 – Temas desarrollados en la clase “Our Toolbox”	30
Ilustración 13 – Evidencia de obtención de parche “Creative Atmosphere Creator”	31
Ilustración 14 – Evidencia de obtención de parche “Agile Thinker”	32
Ilustración 15 – Evidencia de obtención de parche “Active Listener Level 1”	33
Ilustración 16 – Tarea creada para el Trabajo en Progreso del proyecto OIW	35
Ilustración 17 - Porcentaje de Tipos de Prueba	37
Ilustración 18 – Prueba de regresión realizada para el proyecto OIW	37

Ilustración 19 - Porcentaje de Tiempo Dedicado por Proyecto.....	38
Ilustración 20 - Sesiones de Prueba por Proyecto	39
Ilustración 21 - Porcentaje de Errores Encontrados por Clasificación	41
Ilustración 22 - Ejemplo de Documentación de Errores en Proyecto Acera	42
Ilustración 23 – Evidencia de participación en reunión mensual “All Hands”	44
Ilustración 24 – Diagrama de Gantt.....	46

LISTA DE SIGLAS Y GLOSARIO

AA	Acklen Avenue
API	Interfaz de Programación de Aplicaciones
CEO	Director Ejecutivo
CRAI	Centro de Recursos para el Aprendizaje y la Investigación
CTO	Director de Tecnología
GM	Guild Master
GUI	Interfaz Gráfica de Usuario
HU	Hero Unit
OIW	Other Interesting Work
OE	Objetivo Específico
OG	Objetivo General
PP	Práctica Profesional
QA	Control de Calidad
TQL	Tribe/Team Quality Lead
TTL	Tribe/Team Technical Lead
UI	Interfaz Gráfica
UX	Experiencia de Usuario
WIP	Trabajo en Progreso

Ackleners	Empleados de la empresa Acklen Avenue (F. Mezquita, comunicación vía Google Meet, 4 de agosto de 2020)
API	Se define como una especificación de posibles interacciones con un componente de software (Freeman, 2019).
Back-end	Es toda la tecnología necesaria para procesar solicitudes, generar y enviar respuestas al cliente (<i>Back-End Web Architecture</i> , s. f.).
Desarrollador de Software	Se encarga en desarrollar, crear y modificar software general de aplicaciones para computadoras o programas de utilidad especializados (15-1132.00 - <i>Software Developers, Applications</i> , s. f.).
DevOps Engineer	Empleado responsable por la disponibilidad, escalabilidad y salud de los sistemas de la empresa. Son los encargados de desplegar, monitorear y solucionar los problemas que se presenten en los proyectos al trabajar cercanamente con los desarrolladores para coordinar lanzamientos, manejar incidentes y resolver cuestiones de rendimiento (<i>DevOps Engineer</i> , s. f.).
Engagement Manager	Empleado encargado de entrenar tanto al equipo de trabajo como al cliente para ayudar a agregarle valor a los proyectos. Esto requiere crear relaciones, confianza, comprensión de los

fundamentos del proyecto y un liderazgo de servicio que servirá para lograr que los miembros del equipo ejecuten sus roles de la mejor manera (S. Breedlove, comunicación vía Slack, 28 de agosto de 2020).

Front-end Sección gráfica de una aplicación web que le permite al usuario interactuar con los datos mostrados (*What Is a Front-End Developer?*, s. f.).

Guild Nombre utilizado en la empresa para representar un gremio o grupo de empleados que desempeñan el mismo conjunto de responsabilidades. Por ejemplo: Ingenieros en Control de Calidad, Dueños del Producto, Desarrolladores, entre otros (K. Bustillo, comunicación vía Zoom, 31 de julio de 2020).

Guild Master Son empleados que cubren el rol que típicamente cubriría un gerente. Se encargan de que los miembros de su equipo se sientan cómodos, satisfechos y motivados para avanzar en sus carreras. También ayudan a coordinar esfuerzos entre equipos, ayudan con tareas de reclutamiento y son la voz de su equipo con la alta gerencia y en otros roles donde sea necesario (K. Bustillo, comunicación vía Zoom, 31 de julio de 2020).

Heroes Empleados de la empresa Hero Unit (F. Mezquita, comunicación vía Google Meet, 4 de agosto de 2020)

- Kanban Es un enfoque para el cambio de sistemas y procesos incrementales y evolutivos de una organización. Es una forma de organizar el caos que rodea a muchos equipos de trabajo al aclarar la necesidad de priorizar y focalizar. Es una manera de revelar problemas de flujo de trabajo y procesos para solucionarlos con el objetivo de hacer entregas más consistentes al cliente o consumidor (Wester, 2012).
- Product Owner Dueños de Producto. Rol de Scrum que actúa como el punto central de empoderamiento en el liderazgo del producto. Es la única autoridad responsable en decidir qué funciones y funcionalidades crear y el orden a seguir para crearlas (Rubin, 2012, p. 15).
- Programación en Pareja Es un método que se utiliza principalmente en el desarrollo ágil de software y, más concretamente, en la programación extrema (XP). Especifica que siempre haya dos personas trabajando al mismo tiempo en el código y que, en la medida de lo posible, se sientan juntas. Una se encarga de escribir el código y la otra de supervisarlos en tiempo real. Al mismo tiempo, están intercambiando impresiones, debaten problemas, encuentran soluciones y desarrollan ideas creativas (*Pair Programming*, s. f.).

Scrum	Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (<i>Qué es SCRUM</i> , 2008).
Scrum Master	Rol de Scrum que ayuda a todas las personas involucradas a entender y adoptar los valores, principios y prácticas de Scrum (Rubin, 2012, p. 16).
Tribe/Team Quality Lead	Empleado comprometido a ser el mentor del resto de los integrantes del gremio de Ingenieros en Control de Calidad en sus respectivas tribus. Se encarga de eliminar los obstáculos relacionados a QA, identificar áreas de debilidad y colabora con el Guild Master para innovar y mejorar los procesos existentes (K. Bustillo, comunicación vía Google Meet, 13 de agosto de 2020).
Tribe/Team Technical Lead	Empleado encargado de brindar mentoría y asistencia a los desarrolladores líderes de los proyectos y de promover los estándares de calidad dentro de su equipo de trabajo. Colabora con los líderes técnicos en la planificación y mantenimiento de las arquitecturas de software a utilizar en los proyectos internos y

externos, procurando incluir mejoras a las plantillas de los mismos (C. Solórzano, comunicación vía Slack, 28 de agosto de 2020).

Tribe Nombre utilizado en la empresa para representar una tribu o equipo de trabajo encargado de desarrollar los proyectos de los clientes. Por ejemplo: Green Owl, Rogue Squadron y Lionware (K. Bustillo, comunicación vía Zoom, 31 de julio de 2020).

WIP El Trabajo en Progreso se refiere al trabajo que ha sido empezado pero que todavía no ha sido terminado. Durante el desarrollo de productos el Trabajo en Progreso debe ser reconocido y manejado apropiadamente (Rubin, 2012, p. 48).

I. INTRODUCCIÓN

“La Ley de Moore se refiere a una observación hecha por el cofundador de Intel, Gordon Moore, en 1965, en la que descubrió que el número de transistores por pulgada cuadrada en los circuitos integrados se había estado duplicando año tras año desde su invención” (Navas, 2018).

Así como desde hace ya 55 años la sociedad ha experimentado con los frutos de la Ley de Moore, cuyo impacto predomina en la industria de fábrica de hardware principalmente, el mismo impacto se ha visto reflejado en la industria de desarrollo de software, que año tras año, demuestra sus avances con aplicaciones que cada vez se vuelven más complejas y que requieren de bastante tiempo y esfuerzo para ser desarrolladas. Esto conlleva a la creación de cientos o incluso hasta de miles de líneas de código que eventualmente requerirán ser evaluadas y corregidas de ser necesario.

Es aquí donde entra el trabajo de un Ingeniero en Control de Calidad (QA Engineer), cuyo objetivo principal es asistir en el diseño, desarrollo e implementación de diferentes soluciones de software en un ambiente colaborativo y de desarrollo ágil. Su labor también consiste en analizar y comprender el código existente, optimizarlo bajo los estándares de la industria y corregir sus errores en los diferentes ambientes de desarrollo. Todas estas responsabilidades se han convertido en los objetivos que el autor desea lograr durante su práctica profesional en la empresa Acklen Avenue | Hero Unit, con la intención de adquirir experiencia laboral que lo ayudará a desafiar sus conocimientos y a seguir creciendo como profesional en el área de desarrollo de software.

El presente informe contiene una descripción de las generalidades de la empresa, se describe el puesto donde se trabajó y también se comparten los objetivos, tanto el general como los específicos que se desean lograr durante la práctica profesional.

En el marco teórico se mencionan las herramientas utilizadas, los conceptos aprendidos durante el entrenamiento, que a su vez fueron implementados durante la práctica, y los diferentes tipos de pruebas que se realizaron.

La sección de desarrollo muestra las actividades completadas, desde el entrenamiento y crecimiento profesional, hasta el trabajo realizado en un ámbito organizacional y cotidiano, por lo que se entra en detalle para lograr un mejor entendimiento de las asignaciones.

II. GENERALIDADES DE LA EMPRESA

2.1 DESCRIPCIÓN DE LA EMPRESA

Acklen Avenue y su contraparte, Hero Unit, son empresas que buscan construir relaciones de las cuales surge código personalizado que cumple con las expectativas de sus clientes. El trabajo que realizan es 100% remoto pero mantienen su sede principal en Nashville, Tennessee, en los Estados Unidos. La mayoría de sus empleados son de Honduras, aunque actualmente también hay empleados de otras áreas de la región Centroamericana e incluso de países de Sudamérica. Mezquita (2018) escribe en un artículo: Hemos crecido a grandes pasos tomando nuevos desafíos e iniciativas. Sobreparamos la diversión, el empoderamiento, el éxito, la elegancia y la comunidad y nos encontramos regresando a lo que nos define.

Acklen Avenue se define a sí mismo con los siguientes cinco valores:

- Integridad

Haremos lo correcto, aunque duela, al no ser partícipes o apoyar la deshonestidad, corrupción, o injusticia hacia cualquier grupo de personas. Haremos lo que sea mejor para nuestros clientes, incluso si eso significa perder dinero o decir adiós. Seremos honestos en todo lo que hagamos (Mezquita, 2018).

- Maestría

Seremos mejores de lo que fuimos ayer al encontrar nuevas formas para mejorar cada día y al bienvenir mejores formas de hacer las cosas. Empoderaremos a nuestros compañeros de trabajo para que mejoren, facilitando el aprendizaje para aquellos dentro y fuera de nuestra organización. Le daremos la bienvenida a la retroalimentación desde todos los ángulos y siempre demandaremos soluciones elegantes (Mezquita, 2018).

- Calidad

Crearemos productos de alta calidad por dentro y fuera al tomar propiedad del éxito de nuestros clientes. Nos tomaremos el tiempo para hacer un gran trabajo con el fin de ayudar a que nuestros clientes obtengan lo que realmente necesitan, no solo lo que dijeron que necesitaban hace 6 meses. Entregaremos un producto que haga a nuestros clientes felices y nos haga dignos de repetir el negocio (Mezquita, 2018).

- Agilidad

Entregaremos productos valiosos, flexibles y sostenibles lo más seguido posible al asociarnos con nuestros clientes con el fin de maximizar el aprendizaje y el valor del producto utilizando un Enfoque Ágil para el desarrollo de software (Mezquita, 2018).

- Amor

Cuidaremos de nuestros ackleners, tanto profesionalmente como personalmente, al mismo tiempo que cuidamos a nuestros clientes y sus éxitos (Mezquita, 2018).

2.2 DESCRIPCIÓN DEL DEPARTAMENTO

En Acklen Avenue (AA) y Hero Unit (HU) los empleados están categorizados en gremios y como parte de su práctica, el autor pertenece al gremio de los Ingenieros en Control de Calidad. En dicho grupo se trabaja en conjunto a otros, como por ejemplo con los desarrolladores o los Dueños del Producto (Product Owners) y se manejan pruebas exploratorias y automatizadas con el propósito de asegurar la calidad del producto.

2.3 OBJETIVOS DEL PUESTO

2.3.1 OBJETIVO GENERAL

Asistir en el diseño, desarrollo e implementación de diferentes soluciones de software en un ambiente colaborativo y ágil para mantener los estándares de calidad de los proyectos asignados lo más alto posible.

2.3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar nuevas funcionalidades mediante el método de programación en pareja.
- Optimizar el código existente bajo los estándares de la industria y siguiendo las buenas prácticas de programación.
- Desarrollar pruebas automatizadas y manuales para velar por el correcto funcionamiento del proyecto asignado.

III. MARCO TEÓRICO

Para obtener un mejor entendimiento del trabajo realizado y las herramientas que sirvieron de apoyo para el mismo, a continuación se describen términos, recursos y procesos directamente relacionados con las tareas elaboradas durante la práctica profesional.

Acklen Avenue y Hero Unit son empresas 100% remotas y como tales, se respaldan de un conjunto de herramientas virtuales tanto en sus áreas de entrenamiento y desarrollo profesional, así como en sus labores diarias.

La primera herramienta presentada fue durante el entrenamiento y se llama Badgr, la cual es una plataforma utilizada por miles de organizaciones alrededor del mundo para crear ecosistemas de aprendizaje personalizados que apoyan a sus comunidades con credenciales digitales basadas en capacidad y senderos de enseñanza apilables. Dicha herramienta es usada en la empresa para otorgar parches digitales que contribuyen al crecimiento profesional de los empleados. Según Iafrate (2017), un parche digital es un indicador de cumplimiento o competencia que puede ser mostrado, accedido y verificado en línea. El primer entrenamiento que se recibió consistió en una demostración de la correcta navegación de un sendero de enseñanza en Badgr. El lector puede referirse a la sección 4.1.1 del informe para leer más sobre el trabajo realizado.

La siguiente herramienta empleada también fue introducida durante el entrenamiento del autor y se trata del servicio web de Google para asignar tareas, potenciar la colaboración y promover la comunicación entre maestros y estudiantes llamado Classroom. Esta plataforma ayuda a ahorrar tiempo, facilita la creación de clases y mantiene los recursos de manera organizada (DiMaria, 2016). AA y HU hacen uso de este programa para elaborar las clases que comprenden los distintos parches digitales en los senderos de enseñanza de Badgr. Estos se describen más a fondo en la sección 4.1.2.

Sin embargo, Classroom no es el único servicio de Google que la empresa utiliza. A continuación se describen tres herramientas de Google que forman parte del día a día en la organización con sus respectivos usos:

- Gmail: Es un servicio gratis de correo electrónico introducido por Google que permite enviar y recibir correos a través del Internet (*What Is Gmail*, s. f.). Es el medio oficial de comunicación y punto central de información de las actividades que se llevan a cabo en la empresa, no obstante, no es el método de contacto más utilizado por los empleados. Más adelante se hablará sobre este método.
- Google Calendar: Es un calendario web y móvil gratis que te permite controlar tus eventos y compartir tus calendarios con los demás. Es la herramienta ideal para manejar horarios personales y profesionales (Karch, 2019). A través de este servicio se agendan todas las reuniones de la compañía, desde una reunión diaria que puede durar tan poco como 15 minutos, hasta una reunión mensual que puede durar hasta más de 1 hora. Quizás una de las ventajas más grandes de usar este programa sea la facilidad de acceso a los calendarios de los empleados de la empresa. Puedes elegir compartir solo momentos cuando estés ocupado, compartir acceso de solo lectura a detalles de un evento, compartir la habilidad de editar eventos en tu calendario o de manejarlo e invitar a otros (Karch, 2019). En pocas palabras, todos los empleados, en cualquier momento, pueden entrar al calendario de un empleado y ver su disponibilidad de tiempo para agendar una reunión de ser necesario.
- Google Meet: Una vez agendada una reunión, este es uno de los programas que se utiliza para las reuniones oficiales de la empresa. También conocido como Google Hangouts Meet, este servicio está construido para permitirle a docenas de personas unirse a la misma reunión virtual y hablar o compartir videos entre sí desde cualquier lugar con acceso a Internet (John, 2020). Ha sido mediante este programa que se atendió a todas las reuniones que se detallan en la sección 4.1.4 y quizás uno de los beneficios más útiles que incluye sea la capacidad de compartir la pantalla del usuario, lo cual se hizo en todas las reuniones en las que se realizaron pruebas en colaboración con el resto del equipo de QA o con algún otro miembro del equipo de trabajo. Cabe destacar que esta no es la única herramienta que se utilizó para atender a reuniones. Más adelante se hablará sobre la segunda herramienta.

El recurso más utilizado para la comunicación oficial y extraoficial dentro de la organización es un programa llamado Slack. Esta plataforma de mensajería está basada en canales y con ella, un grupo de personas puede trabajar de manera más eficiente, conectar herramientas de software y servicios, y encontrar información rápidamente para realizar un mejor trabajo, todo dentro de un ambiente seguro y de grado empresarial (*What Is Slack?*, s. f.). Se manejan dos espacios de trabajo principales, uno para Acklen Avenue y uno para Hero Unit. Un espacio de trabajo en Slack está compuesto por canales, donde miembros de un equipo pueden comunicarse y trabajar juntos (*What Is Slack?*, s. f.). Al formar parte de proyectos relacionados a ambas empresas, el autor pertenece a los dos espacios de trabajo, y fue a través de ellos que la mayor parte de la comunicación durante toda la práctica profesional tomó lugar.

En cuanto a cómo se asignaron las tareas que se realizaron, es aquí donde entra otro programa de bastante uso por la compañía: Trello, el cual es una herramienta de colaboración que organiza proyectos en muros. A simple vista, Trello muestra en qué se está trabajando, quién está trabajando, y dónde se encuentra ese trabajo dentro de un proceso (*What is Trello?*, 2017). A este muro también se le conoce como muro Kanban y es sobre él que se asignan las tareas a los diferentes miembros del equipo. Para el caso particular de un Ingeniero en Control de Calidad, sus tareas son asignadas en relación a las tareas de un Desarrollador de Software. Una vez que este las termina, las tareas se mueven a la lista (o columna) de Ready to Test (Listo para Probar). Una lista es una colección de tarjetas (o tareas) ordenadas verticalmente. Pueden representar una colección de ideas, recordatorios, o las diferentes etapas del flujo de trabajo (*Adding lists to a board*, 2019). El Ingeniero en Control de Calidad luego toma esa tarjeta, la prueba, y la vuelve a mover a través del muro, siguiendo su camino dentro del Trabajo en Progreso. Esto sucede para cada ambiente de pruebas que existe, los cuales varían de proyecto a proyecto.

En la Ilustración 1 se puede observar un ejemplo sencillo de un muro Kanban.

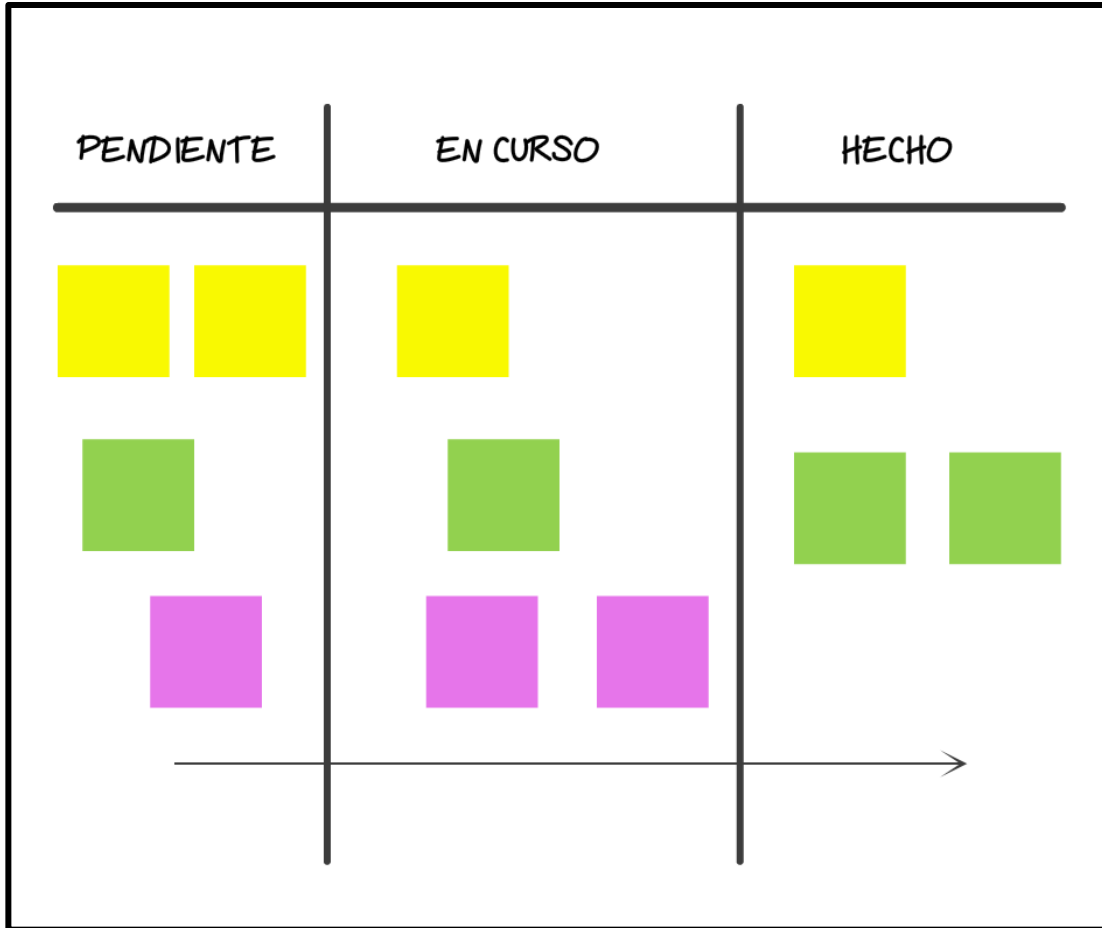


Ilustración 1 – Ejemplo de Muro Kanban

Fuente: (Javier, 2017)

Al finalizar una jornada de trabajo, AA y HU requieren que cada uno de sus empleados registre sus horas trabajadas durante el día. Para realizar esto se hace uso de Peeplo, un programa desarrollado internamente para llevar una bitácora de los intervalos de tiempo en los que se realizó trabajo para la empresa. Anteriormente se utilizaba Jibble, una solución para el manejo de asistencia y seguimiento temporal basada en la nube diseñada para ayudar a negocios y organizaciones a maximizar el tiempo de sus empleados y asegurarse que les paguen por el tiempo real que han trabajado (Gilbert, 2020). Sin embargo, a diferencia de Jibble que requiere de dos pasos para registrar una bitácora, Peeplo solo requiere de uno, puede hacer todo lo que hace Jibble y nuevamente, fue desarrollado como un proyecto interno de la empresa. Este

programa se usó desde que fue introducido durante el entrenamiento y se siguió usando por el resto de la práctica profesional.

Ahora se hablará de Zoom, la segunda herramienta utilizada para atender a reuniones de la compañía, específicamente a las reuniones que tomaron lugar durante el entrenamiento y las reuniones mensuales que se describen en la sección 4.1.5. Zoom ayuda a los negocios y organizaciones a juntar a sus equipos en un ambiente sin fricción para completar más trabajo. Su fácil y confiable plataforma en la nube para compartir video, voz, contenido y conversaciones funciona en dispositivos móviles, ordenadores de escritorio y sistemas de sala (*About Zoom*, s. f.). Este programa se utilizó durante el entrenamiento porque en ese momento todavía no se contaba con un correo empresarial, y se utilizó en las reuniones mensuales por la sencilla razón que con el plan que maneja la empresa, se pueden programar llamadas con más de 100 participantes.

Para darle fin a la descripción de las herramientas utilizadas a lo largo de la práctica, a continuación se mencionan tres programas de menor relevancia, pero que también se involucraron en las labores realizadas:

- Novabench: Es un indicador gratuito que evalúa el rendimiento de una computadora (*Novabench*, s. f.). Este programa fue utilizado para la clase “Welcome to Our Family” (ver sección 4.1.2.1) con el propósito de corroborar los requerimientos mínimos del sistema de la computadora que se utilizaría durante las próximas 22 semanas.
- GitHub: Es una plataforma de alojamiento de código para control de versiones y colaboración (*Hello World – GitHub Guides*, 2020). En vista que el puesto del autor dentro de la compañía no requiere de programación, GitHub ha sido una herramienta de muy poco uso, no obstante, se recibieron talleres de GitHub y se requirió crear una cuenta en caso de que fuera necesario utilizarla para alguna tarea.
- GitLab: Es una plataforma completamente para DevOps, presentada como una sola aplicación. Sirve para crear un flujo de trabajo de software simple y para liberar a las organizaciones de las restricciones de una cadena de herramientas entrelazada (*What Is GitLab?*, s. f.). Este programa sirvió de apoyo para las

reuniones de pruebas en colaboración con los DevOps Engineers y para verificar el estado de los back-ends de los proyectos asignados.

A lo largo del entrenamiento se introdujeron ciertos conceptos, algunos con el propósito de ser el contenido de una evaluación y otros meramente con fines didácticos. A pesar de que se tenía conocimiento sobre varios de ellos, para algunos se requirió investigar y la empresa esperó que fueran implementados a lo largo de la práctica profesional. A continuación se describen estos conceptos, así como su utilidad en las actividades realizadas:

- Metodologías ágiles

Antes de hablar sobre las metodologías, es importante saber que significa ser ágil. Según Reyes (s.f.), ser ágil significa trabajar de una manera ligera y altamente responsable para entregar un producto o servicio tal y como el cliente lo quiere y al momento que el cliente lo necesita. No existe una sola forma de ser ágil, sin embargo, este concepto nace de dos ideas que son aceptadas por la mayoría de las personas cuando se trata del desarrollo ágil de software:

1. El Manifiesto Ágil

Según Crispin & Gregory (2009), usando los valores del Manifiesto para guiarnos, nos esforzamos en entregar pequeños pedazos de valor empresarial en ciclos de lanzamiento extremadamente cortos (p. 3). En la Ilustración 2 se pueden observar los valores del Manifiesto Ágil para el desarrollo de software.

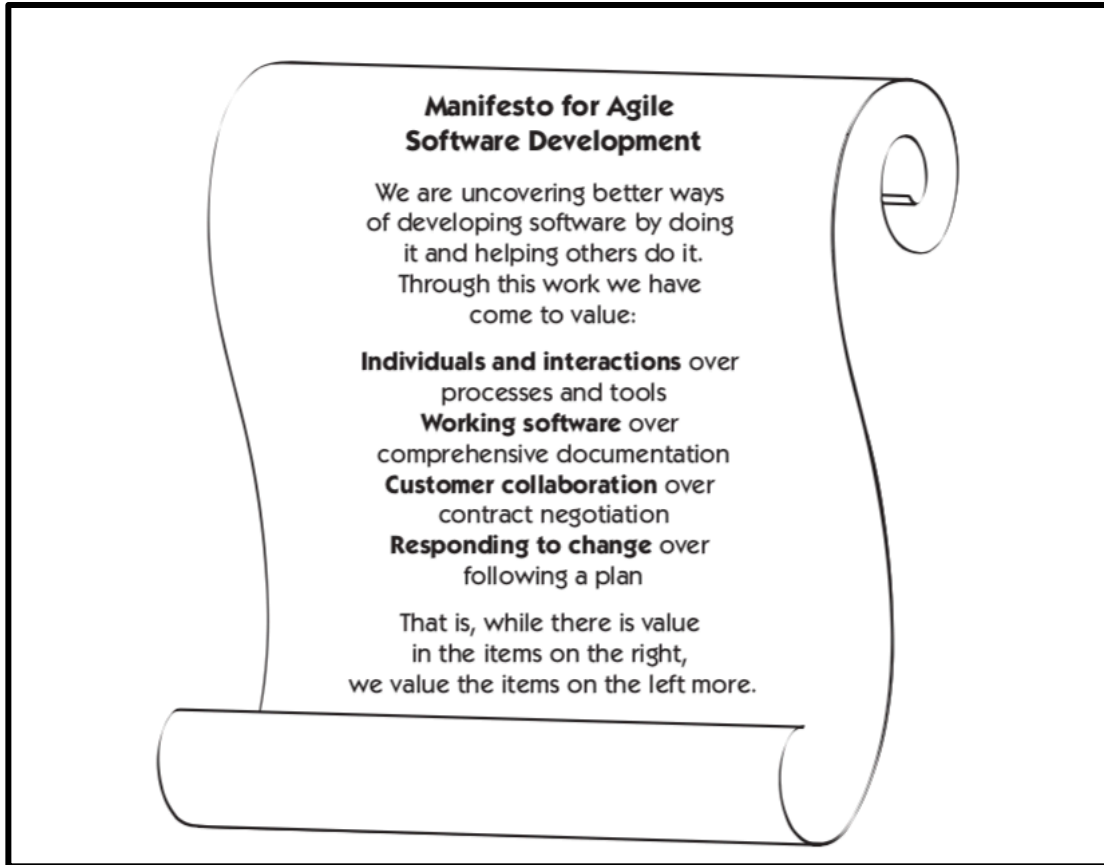


Ilustración 2 – Manifiesto Ágil

Fuente: (Crispin & Gregory, 2009, p. 4)

2. Los 12 Principios del Desarrollo Ágil de Software

En la Ilustración 3 se pueden observar los 12 Principios del Desarrollo Ágil de Software.

The following 12 Principles are based on the [Agile Manifesto](#).

1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7	Working software is the primary measure of progress.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9	Continuous attention to technical excellence and good design enhances agility.
4	Business people and developers must work together daily throughout the project.	10	Simplicity—the art of maximizing the amount of work not done—is essential.
5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11	The best architectures, requirements, and designs emerge from self-organizing teams.
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Ilustración 3 – Los 12 Principios del Desarrollo Ágil de Software

Fuente: (12 Principles Behind the Agile Manifesto, 2015)

Las metodologías ágiles ponen en práctica los principios y valores de las dos ideas previamente mencionadas. “Poniendo un ejemplo, en el marco de trabajo Scrum, el proyecto se divide en pequeñas partes que tienen que completarse y entregarse en plazos cortos, llamados sprints” (Tena, 2020). Los sprints son “timeboxed”, lo que significa que siempre tienen fechas fijas de inicio y de fin y generalmente todos deben tener la misma duración (Rubin, 2012, p. 20). El objetivo de estos sprints es que al completarlos, se logre entregar un avance significativo al cliente, cumpliendo así con el valor “Working software over comprehensive documentation” (Software funcional sobre documentación comprehensiva) del Manifiesto Ágil. Lo mismo debe suceder para cada método y proceso que pertenezca a una metodología ágil. En Acklen

Avenue y Hero Unit, la metodología que se aplica es Kanban para el desarrollo de software, mezclada con algunos elementos de Scrum.

- Mejora continua

“El objetivo de la mejora continua es mejorar la competitividad de las empresas (en cualquiera de sus áreas) a través de la productividad de una manera permanente y sostenible en el tiempo” (*¿Qué es la Mejora Continua?, 2014*). Ha sido gracias a la mejora continua que la empresa ha logrado crecer tanto en los últimos años. Pasó de ser una empresa que trabajaba de manera desorganizada y que no brindaba satisfacción a sus clientes y empleados a ser una empresa que constantemente está entregando productos de calidad y que mantiene un ambiente laboral bastante agradable. Una de las formas en las que se implementa la mejora continua en la empresa es al permitirle a todo el personal dedicar hasta 1 hora al día de entrenamiento en cualquier tema que vaya a ser de beneficio profesional. Fue durante este espacio de tiempo que se trabajó en las actividades descritas en la sección 4.1.3.

- Manejo de deuda técnica

En 1992, Ward Cunningham usó la metáfora de deuda técnica para explicarle a su equipo de negocios por qué crear software rápido para obtener retroalimentación es algo bueno. Hoy en día, la deuda técnica se refiere tanto a los atajos que intencionalmente se toman, así como a todas las malas prácticas que plagan los sistemas de software (Rubin, 2012, p. 139). Estas prácticas incluyen: malos diseños, defectos, insuficiente cobertura de pruebas, exceso de pruebas manuales, entre otras.

Las siguientes consecuencias podrían producirse a causa de no manejar la deuda técnica:

1. Punto de inflexión impredecible

En algún momento, la deuda técnica llega a un punto de inflexión y se vuelve inmanejable o caótica (Rubin, 2012, p. 142). Es decir, entre más problemas se

acumulen, más impredecible se vuelve el momento en el que todo puede llegar a fallar en un proyecto.

2. Tiempo de entrega incrementado

Esto quiere decir que entre más tiempo se ha desplazado para trabajar en un problema en el futuro, más largo se volverá el tiempo de entrega del producto, considerando que primero se tendrán que solventar los problemas antes de hacer la entrega del mismo.

3. Número significativo de defectos

Los fallos se convierten en una interrupción mayor del flujo normal de incremento al valor del producto (Rubin, 2012, p. 142). Esto significa que entre más errores haya, estos se pueden volver más difíciles y complejos, cuando en un inicio pudieron haber sido mucho más fáciles de solucionar.

4. Atrofia del producto

Si se llega a un punto en el que no se logra agregarle más valor al producto por estar solucionando problemas pasados, este cada se volverá menos atractivo para los consumidores actuales y potenciales, causando una atrofia del mismo como resultado.

5. Desempeño insuficiente

Tristemente, mientras la deuda técnica incrementa, se empieza a esperar cada vez menos desempeño en desarrollo y por ende, se reducen las expectativas de lo que es posible realizar (Rubin, 2012, p. 143).

Luego de haber sido introducido a este concepto y una vez comprendiendo la severidad a la que este puede llegar, se investigaron las siguientes formas para manejar la deuda técnica y se procuró implementarlas a lo largo de la práctica profesional:

1. Usar buenas prácticas técnicas

A pesar de que Scrum no define formalmente prácticas técnicas, cada equipo de Scrum exitoso que se ha visto emplea prácticas tales como diseños simples, desarrollo en base a pruebas, integración continua, pruebas automatizadas, entre más (Rubin, 2012, p. 149). Si bien es cierto, no todas estas prácticas forman parte del alcance de los proyectos asignados, sin embargo conocer sobre ellas sirvió como un excelente punto de partida y se aplicaron en la medida de lo posible.

2. Usar una buena definición de terminado

Según Rubin (2012), conceptualmente, la definición de terminado es una lista de comprobación de los tipos de trabajo que el equipo espera se completen exitosamente antes de declarar que ese trabajo es potencialmente entregable (p. 74). Es decir, entre más aspectos técnicos abarque la definición de terminado, será menos probable que se acumule más deuda técnica. Para hacer esto, se siguieron todos los formatos de la empresa para documentar errores y se siguieron todos los lineamientos al momento de ejecutar pruebas en base a los requerimientos técnicos realizados por los Desarrolladores de Software.

- Dar un paso adicional (going the extra mile)

Aquellos que están dispuestos a dar un paso adicional son aquellos que demuestran un espíritu de excelencia, lo que conlleva a éxito en todos los aspectos de la vida y de los negocios (Meltzer, 2018). Este concepto fue introducido en la clase "Culture Champion" (sección 4.1.2.4) y desde entonces, se hizo el esfuerzo por implementarlo en todos los aspectos de la práctica, desde mantener una mentalidad positiva en cada labor hecha para la empresa, hasta buscar aprender algo nuevo todos los días, independientemente de la carga de trabajo.

- Servicio al cliente vs experiencia del cliente

La diferencia entre servicio al cliente y experiencia del cliente es que mientras el primero es una pieza del rompecabezas, enfocado en la interacción humana y en apoyar directamente a los clientes, el segundo es la suma del viaje completo de un cliente por un negocio (Blackstock, 2018). Este concepto fue introducido en la clase "Introduction to Client Expectations" (sección 4.1.2.6) y a pesar de que se tuvo poca

interacción con un cliente externo, se aprovechó cada reunión para aprender sobre un escenario de trabajo real y se participó en dichas reuniones cuando fue requerido.

- Método de resolución de problemas A3

Es una técnica que fue creada en Japón por Toyota como parte de su sistema de producción que se asocia con la manufactura lean y es utilizada como un esfuerzo de mejora continua en varios lugares de trabajo (Dalto, 2020). Provee un método lógico para resolver un problema y comunicarlo con los demás a través de los siguientes pasos:

1. Antecedentes: se identifica y se describe el problema.
2. Situación actual: se explica la situación actual y se provee contexto.
3. Objetivo o meta: se definen las metas a las que se quiere llegar. Es posible regresar a este paso si se descubren más metas.
4. Análisis de causa: se busca la raíz del problema.
5. Contramedidas: se proponen soluciones que potencialmente podrían ayudar a llegar a la meta.
6. Plan de implementación: se elabora un plan que detalla paso a paso cómo se implementará cada contramedida.
7. Implementación: se ejecuta el plan.
8. Verificación: se verifica que el plan de hecho haya funcionado. En caso de no hacerlo, se elabora un nuevo plan, se ejecuta, y se vuelve a verificar. Este proceso se repite hasta encontrar una contramedida que funcione.
9. Seguimiento: si el plan funciona exitosamente, se comunican los resultados con el resto de compañía con el fin de que las personas tengan la oportunidad de probarlo ellas mismas y se convierta en una norma.

Se elaboró un plan de mejora continua siguiendo el método de resolución de problemas A3 para la clase "Remote Professional" (sección 4.1.2.7) como parte de una tarea del entrenamiento.

Finalmente, el deber más importante de un Ingeniero en Control de Calidad son las pruebas que realiza para apoyar al equipo de desarrollo. Existen distintas pruebas y cada una de ellas conlleva su propio conjunto de operaciones. Para entender las

asignaciones primordiales de un Ingeniero en QA se necesita ver algunos tipos de pruebas que este ejecuta o analiza:

- Pruebas en Pareja: Es una técnica de pruebas de software que puede ser realizada por los evaluadores, desarrolladores y analistas de negocios. Como el nombre sugiere, dos personas se juntan, una para probar y otra para monitorear y documentar los resultados. Las pruebas en pareja pueden ser ejecutadas en combinaciones tales como evaluador-desarrollador, evaluador-analista de negocios o desarrollador-analista de negocios. Combinar a evaluadores y desarrolladores ayuda a detectar errores rápido, identificar la causa, arreglarlos y probarlos (Ghahrai, 2019).
- Pruebas Exploratorias: Son un tipo informal de pruebas que se realiza para aprender sobre el software mientras se buscan errores o comportamientos de la aplicación que no parezcan tan obvios. Usualmente las hacen los evaluadores pero también las pueden hacer otras partes interesadas con el objetivo de aprender sobre el software, tales como los analistas de negocio, desarrolladores, usuarios finales, entre más (Ghahrai, 2019).
- Pruebas de Regresión: Es un tipo de prueba de software que se lleva a cabo por el equipo de control de calidad como pruebas de regresión funcional y los desarrolladores como pruebas de regresión de unidad. El objetivo de estas pruebas es encontrar los defectos que se introdujeron para corregir los defectos o introducir nuevas características. Las pruebas de regresión son candidatos ideales para la automatización (Ghahrai, 2019).
- Pruebas de Caja Negra: Es un método de prueba de software en el que no se requiere conocer la codificación o la estructura interna del software. Este método se basa en probar el software con varias entradas y validar los resultados obtenidos contra los resultados esperados (Ghahrai, 2019).
- Pruebas Funcionales: Se centran en la comprobación del software contra el documento de diseño, los casos de uso y el documento de requisitos. La prueba funcional es un tipo de prueba de caja negra y no requiere funcionamiento interno del software (Ghahrai, 2019).
- Pruebas de Integración: Son uno de los tipos más comunes e importantes de pruebas de software. Una vez que las unidades o componentes individuales

son probados por los desarrolladores como trabajo, entonces el equipo de evaluación ejecutará pruebas que comprobarán la conectividad entre estas unidades/componentes o unidades/componentes múltiples (Ghahrai, 2019).

- Pruebas de API: Es un tipo de prueba similar a las pruebas de unidad. Cada API del software es probada según su especificación. Las pruebas son realizadas en su mayoría por el equipo de pruebas a no ser que la API sea muy compleja o necesite código extensivo. Estas pruebas requieren entender la funcionalidad del API y poseer buenas habilidades de codificación (Ghahrai, 2019).
- Pruebas de GUI: Este tipo de prueba está enfocada en corroborar que la Interfaz Gráfica de Usuario cumpla con los requerimientos mencionados en los prototipos y documentos de diseño. Por ejemplo, revisar la longitud y capacidad de los campos de un formulario, el tipo del campo, si algunos campos pueden ser mostrados como un menú desplegable, entre más (Ghahrai, 2019).

Principalmente se manejaron las siguientes pruebas durante la práctica:

- Pruebas en Pareja: Al colaborar con un Desarrollador de Software, un DevOps Engineer o algún otro miembro del equipo de QA para agilizar el proceso de pruebas o para profundizar en algún aspecto técnico del software.
- Pruebas Exploratorias: Al navegar libremente por el sitio o aplicación a probar con el fin de conocerlo e ir descubriendo errores en la medida de lo posible. También fue un método bastante útil para aprender sobre los proyectos asignados.
- Pruebas de Regresión: Al repasar cada una de las funcionalidades existentes de un proyecto, así como las nuevas funcionalidades que se fueron agregando. Los Desarrolladores de Software y los TTLs indicaron cuándo realizar estas pruebas.
- Pruebas Funcionales: Al apoyar a otros miembros del equipo de QA en proyectos existentes y siguiendo documentos preestablecidos con los requerimientos que estos debían tener.

- Pruebas de GUI: Al colaborar con los Diseñadores de UI/UX y apegarse a los diseños aprobados por los clientes. También se sugirieron cambios en la Interfaz Gráfica de Usuario de proyectos existentes cuando se consideró apropiado.

IV. DESARROLLO

4.1 DESCRIPCIÓN DEL TRABAJO REALIZADO

4.1.1 ENTRENAMIENTO: PATHWAY ORIENTATION

Como una orientación al entrenamiento que posteriormente se realizaría, se recibió una demostración de la correcta navegación de un sendero de aprendizaje (pathway) de la plataforma Badgr. Dicha demostración consistió en dos videos instructivos que fueron seguidos de la creación de una carpeta en Google Drive que serviría de apoyo para el resto del entrenamiento.

En la Ilustración 4 se puede observar la captura de pantalla de uno de los videos instructivos impartido por el Director de Tecnología de la empresa, Byron Sommardahl.

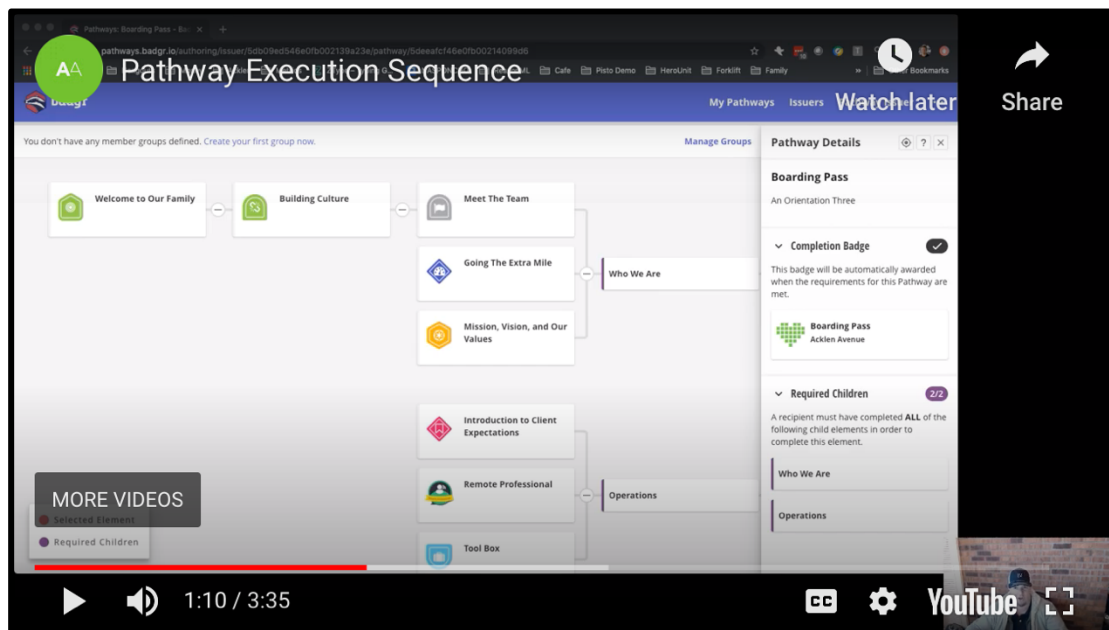


Ilustración 4 – Evidencia de video instructivo en Pathway Orientation

Fuente: (Byron_Sommardahl, 2019).

4.1.2 ENTRENAMIENTO: BOARDING PASS PATHWAY

En el sendero de aprendizaje como pase de abordaje a la empresa (Boarding Pass Pathway) se concentró la mayor parte de la capacitación recibida. Dicha capacitación consistió en 9 clases en la plataforma Google Classroom y culminó con la evaluación para la obtención del parche digital llamado "Creador de una Atmósfera Creativa" ("Creative Atmosphere Creator"). Todas las clases, sus contenidos, tareas, investigaciones y la evaluación final se detallan a continuación.

4.1.2.1 Clase "*Welcome to Our Family*"

En esta clase se dio la bienvenida a la empresa, se comentó la razón por la que Acklen Avenue ha decidido confiar en Honduras como fuente de trabajo y luego se mencionaron conceptos que son inculcados en la empresa: Mejora continua, manejo de la deuda técnica y software de calidad.

La siguiente parte de la clase se basó en realizar una serie de tareas con el objetivo de corroborar que se tuvieran los requerimientos mínimos del sistema en la computadora para llevar a cabo el trabajo de la mejor manera. Para dichas tareas se requirió el uso del programa Novabench, así como la elaboración de dos videos que demostraron la calidad del equipo audiovisual que se estaría usando en la práctica profesional para la reuniones virtuales.

Finalmente, se compartió la lista de beneficios de los cuales gozan los empleados de AA y HU, se solicitó subir una foto de primer plano en todas las herramientas que utiliza la empresa, se realizó una pequeña tarea para configurar los correos electrónicos de trabajo y se hizo una presentación frente a la organización y frente al Director Ejecutivo, Colin Callahan, a través de la plataforma Slack.

En la Ilustración 5 se puede observar una captura de pantalla de uno de los videos elaborados por el autor para esta clase.

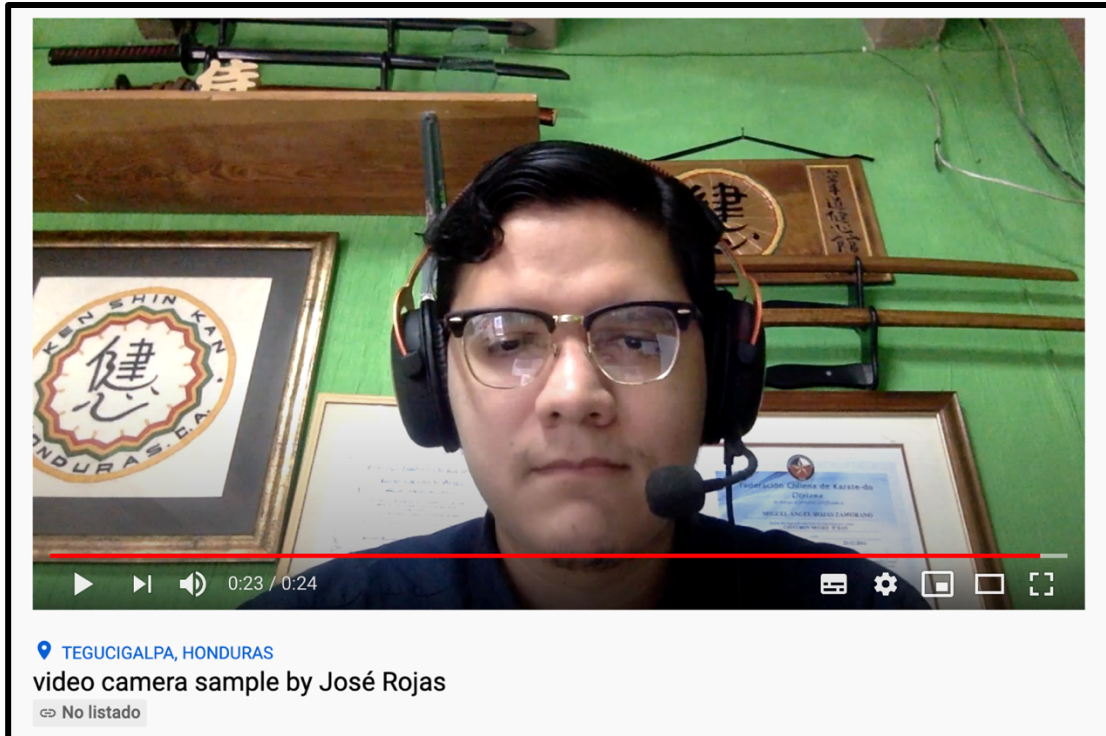


Ilustración 5 – Evidencia de video elaborado para clase “Welcome to Our Family”

Fuente: (Elaboración Propia)

4.1.2.2 Clase “Building Community”

Esta clase se enfocó en enseñar la cultura organizacional de la empresa al presentar un ejemplo de las reuniones mensuales que suceden con todos los empleados y personal administrativo, conocidas como All Hands. También se compartieron imágenes de reuniones de Acción de Gracias que la empresa ha tenido en años anteriores, así como de otras actividades en las que se ha reunido al personal, tales como campamentos y patrocinios. La clase terminó con un extenso video que abordaba un concepto que vendría siendo muy importante para el desarrollo dentro de los proyectos: La metodología ágil.

En la Ilustración 6 se puede observar una captura de pantalla de los temas desarrollados en esta clase.

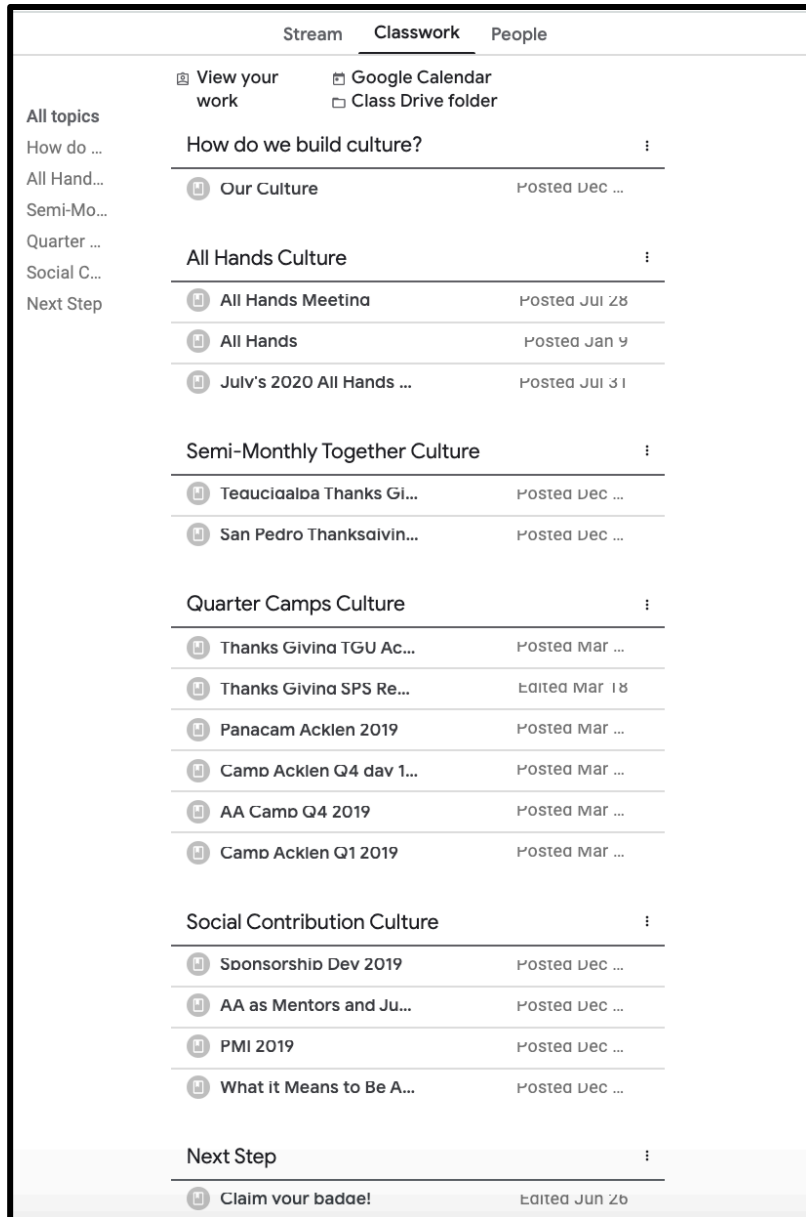


Ilustración 6 – Temas desarrollados en la clase “Building Community”

Fuente: (Garcia, Cristian & Sommardahl, Byron, 2019a)

4.1.2.3 Clase “Meet the Team”

En esta asignatura se conocieron a los integrantes de los diferentes gremios (o equipos) que existen en la empresa, tales como: People Operations (Recursos Humanos), Dueños de Producto, Desarrolladores de Software, Diseñadores de Interfaz Gráfica y Experiencia de Usuario, Ingenieros en Control de Calidad, entre otros.

En la Ilustración 7 se puede observar una captura de pantalla de los temas desarrollados en esta clase.

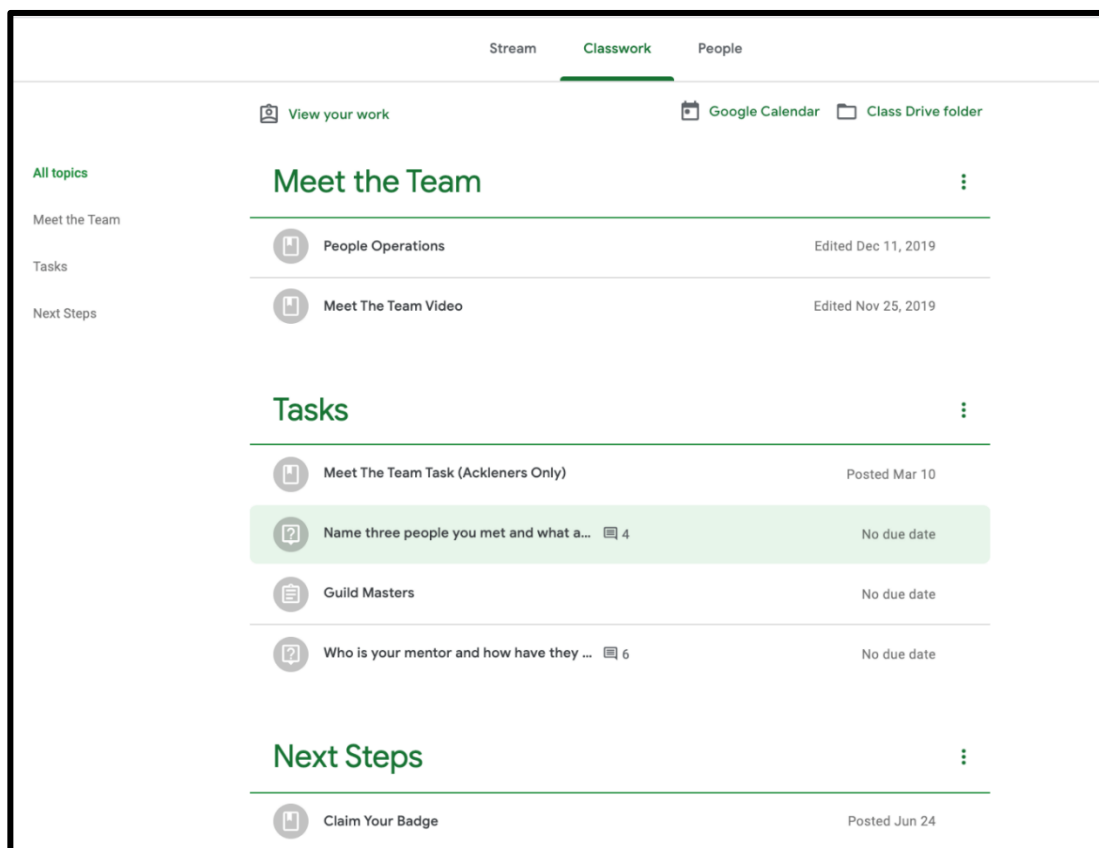


Ilustración 7 – Temas desarrollados en la clase “Meet the Team”

Fuente: (Garcia, Cristian & Somnardahl, Byron, 2019c)

4.1.2.4 Clase “Culture Champion”

Esta fue quizás la clase más corta de todo el sendero de aprendizaje y únicamente consistió en la enseñanza de un concepto que la empresa espera que el autor ponga en práctica: Dar un paso adicional (going the extra mile).

En la Ilustración 8 se puede observar una captura de pantalla de los temas desarrollados en esta clase.

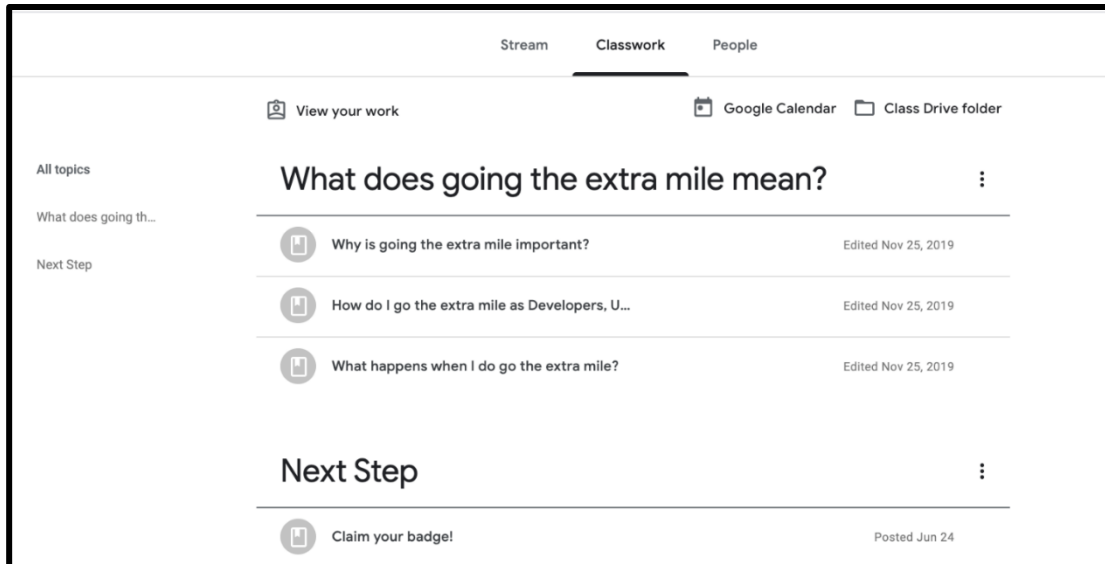


Ilustración 8 – Temas desarrollados en la clase “Culture Champion”

Fuente: (Garcia, Cristian & Sommardahl, Byron, 2019b)

4.1.2.5 Clase “Our Vision and Values”

En esta clase se aprendió la visión y los valores de la empresa a través de un documento redactado por la Directora de Operaciones de Negocio y Marketing y jefa inmediata del autor, Fernanda Mezquita, y a través de cinco videos, uno para cada valor, los cuales son: Calidad, Integridad, Amor, Agilidad y Maestría.

En la Ilustración 9 se puede observar una captura de pantalla de los temas desarrollados en esta clase.

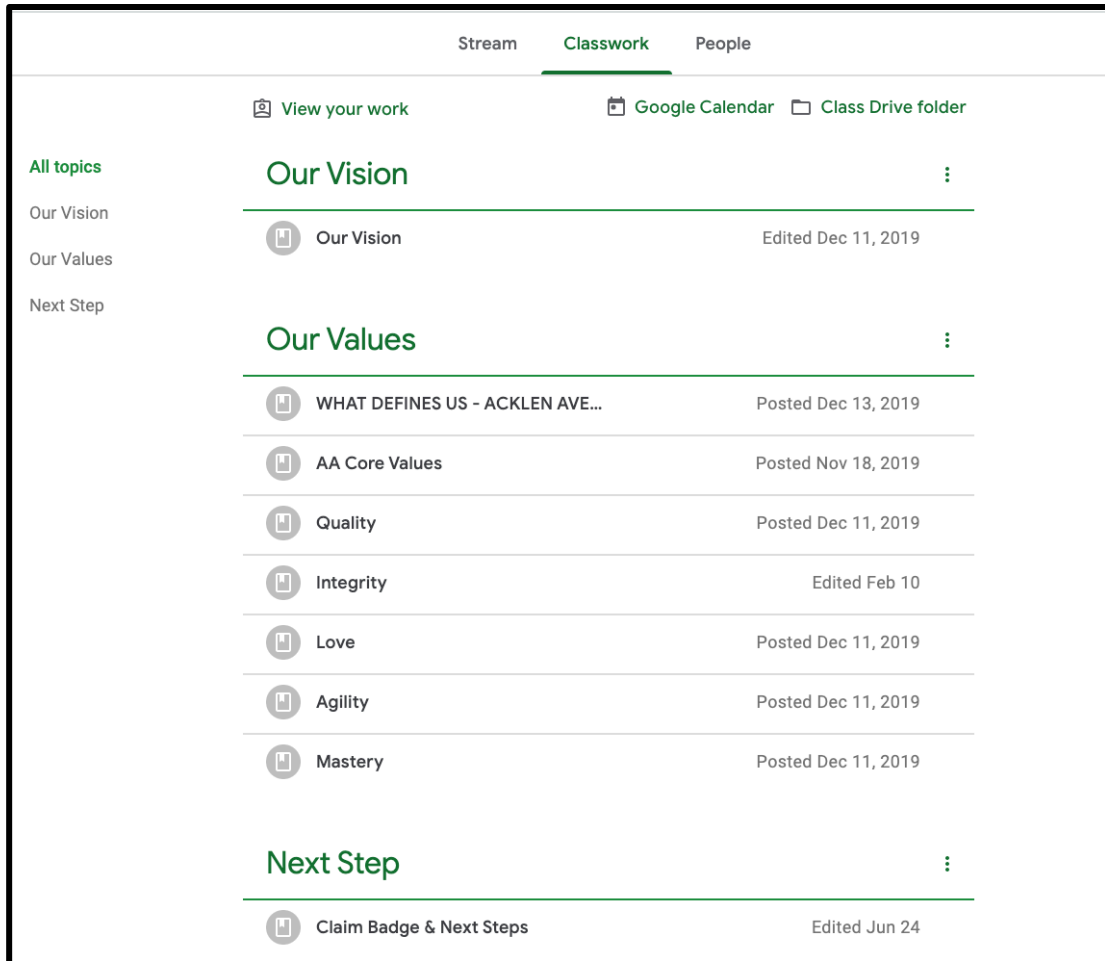


Ilustración 9 – Temas desarrollados en la clase “Our Vision and Values”

Fuente: (Garcia, Cristian et al., 2019b)

4.1.2.6 Clase “Introduction to Client Expectations”

Esta clase se enfocó en enseñar una serie de conceptos relacionados a la interacción con los clientes mediante videos instructivos, lecturas y recomendaciones, con el objetivo de mejorar las habilidades en las siguientes áreas: servicio al cliente, experiencia del cliente, etiqueta para reuniones remotas, cómo desarrollar buenas relaciones, entre más. Para finalizar se asignó una tarea que consistió en redactar consejos para construir buenas relaciones con los clientes, los cuales se pueden observar en la Ilustración 10.



Jose Carlos Jul 29

1. Treat your clients as you would treat a loved one. Try to understand what they are trying to say rather than just giving them a hollow response.
2. Never, under any circumstance, engage in a verbal altercation with your clients. Even if they are disrespectful to you, be the better person and try to defuse the situation.
3. This probably applies more to face-to-face interactions but still, just smile more. A quick smile not only presents a kind-hearted demeanor, but it also has shown to improve our own mental state even if forced to do so.
4. Be proactive, but be smart about it. When given the chance to do more than what was asked of you, first analyze the benefit it would have on your client and then decide on taking action. Many people like when you take the extra mile, but others may not share the same feeling. So as with everything, think before you act.

Ilustración 10 – Tarea para la clase “Introduction to Client Expectations”

Fuente: (Elaboración Propia)

4.1.2.7 Clase “Remote Professional”

Esta clase fue una de las más extensas del sendero de aprendizaje y constó de una serie de lecturas, videos, tareas individuales y colaborativas y configuraciones de herramientas que utiliza la empresa.

La primera parte comprendió consejos para el manejo del tiempo al trabajar remotamente. Luego se abarcaron los conceptos de mejora continua y método de resolución de problemas A3, para los cuales se pidió investigar y redactar documentos que ejemplificaran el uso de los mismos.

La segunda parte requirió la creación de un plan para demostrar presencia en el trabajo y otro para demostrar el progreso que se desea realizar dentro de la empresa.

En la tercera y última parte se volvió a mencionar la etiqueta de una reunión remota y se solicitó la colaboración del autor con un compañero de trabajo para elaborar un video ejemplificándola. Finalmente, se realizaron dos trabajos para configurar las herramientas Zoom y Google Calendar, las cuales vendrían siendo de mucho uso durante la práctica profesional.

En la Ilustración 11 se puede observar el plan para demostrar presencia en el trabajo que fue elaborado para esta clase.

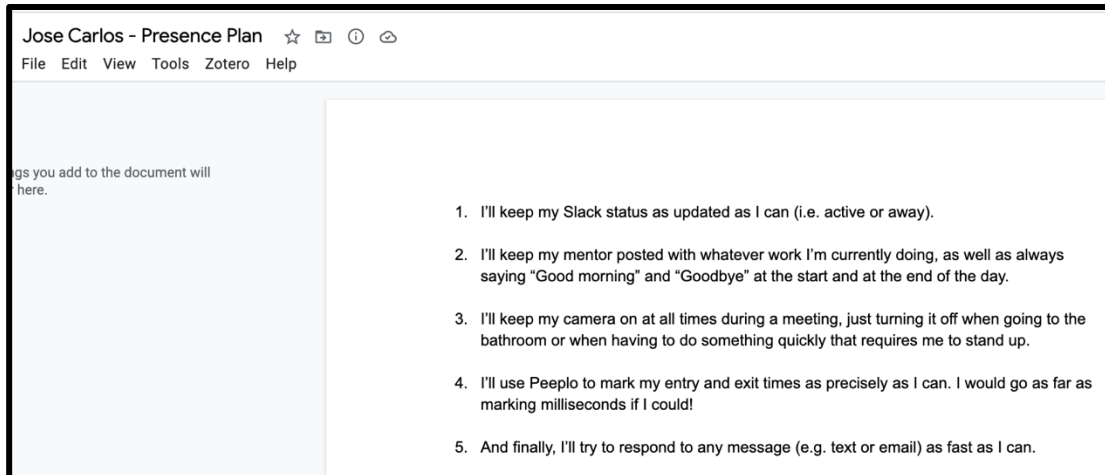


Ilustración 11 – Plan de presencia elaborado para la clase “Remote Professional”

Fuente: (Elaboración Propia)

4.1.2.8 Clase “Our Toolbox”

En esta asignatura se terminaron de configurar el resto de las herramientas que más se utilizan en la empresa y por ende, que se utilizaron en la práctica, las cuales son: Gmail, Google Calendar, Trello, Slack, GitHub y Peeplo. También se informaron los canales apropiados para la comunicación de los diferentes gremios de la organización y por último se compartió la póliza de sueldos y la fórmula utilizada para calcularlos. Se le recuerda al lector que a pesar de haber sido expuesto a esta información, el autor no ha gozado ni gozará de sueldo durante la práctica profesional.

En la Ilustración 12 se puede observar una captura de pantalla de los temas desarrollados en esta clase.

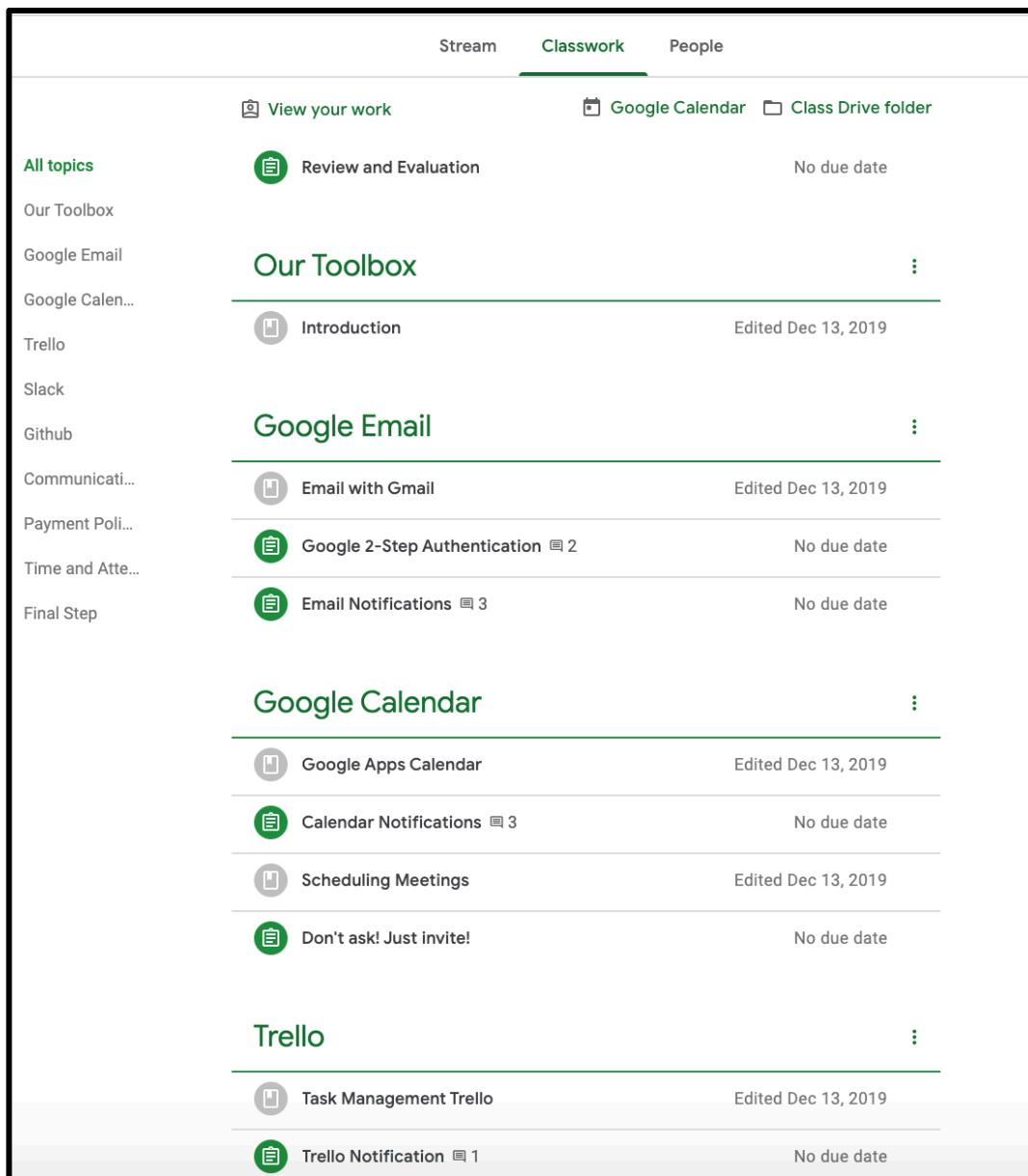


Ilustración 12 – Temas desarrollados en la clase “Our Toolbox”

Fuente: (Garcia, Cristian et al., 2019a)

4.1.2.9 Clase “Dojo Orientation”

En esta, la última clase del sendero de aprendizaje, se aprendió sobre el sistema de evaluación y superación profesional que la empresa tiene implementado que toma inspiración en las artes marciales y se asignó una mentora, cuyo nombre es Kerlyn Bustillo, que también es la GM del gremio de Ingenieros en Control de Calidad. También se solicitó agregar una extensión al navegador de Chrome para facilitar la

obtención de los parches digitales otorgados para todas las clases completadas del sendero y finalmente se realizó una prueba de los temas vistos en esta clase, que fue aprobada con 8 de 10.

4.1.2.10 Evaluación para parche "Creative Atmosphere Creator"

El último parche digital que se obtuvo como parte del entrenamiento fue el "Creador de una Atmósfera Creativa", el cual consistió en una evaluación que requirió investigar y estudiar los siguientes temas: Cómo crear una atmósfera de creatividad, cómo ser más creativos en el trabajo, cinco elementos de una cultura creativa, entre más. La evaluación fue realizada por el Coordinador de Carreras, Cristian García, y se completó exitosamente.

En la Ilustración 13 se puede observar una captura de pantalla del parche digital obtenido luego de aprobar la evaluación.

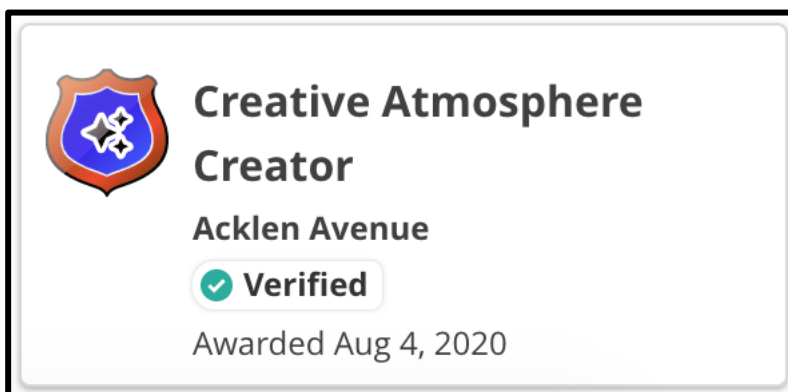


Ilustración 13 – Evidencia de obtención de parche "Creative Atmosphere Creator"

Fuente: (*Atmosphere of Creativity*, 2019)

4.1.3 DESARROLLO PROFESIONAL: ASSOCIATIVE QUALITY ENGINEER I PATHWAY

A partir de esta sección se detalla el trabajo realizado en el segundo sendero de aprendizaje en el que se participó, con la diferencia que este deja de estar relacionado a la capacitación del autor y da inicio a su desarrollo profesional en la empresa como un Ingeniero en Control de Calidad.

4.1.3.1 Evaluación para parche "Agile Thinker"

De manera muy similar a la evaluación del parche “Creador de una Atmósfera Creativa”, para esta también se requirió investigar y estudiar una lista de temas, los cuales fueron: Roles, actividades y artefactos de Scrum, time-boxing, sprints, spikes, definición de terminado y Kanban. La evaluación fue realizada por el GM del gremio de Desarrolladores de Software, Yamil Gonzales, y fue completada exitosamente.

En la Ilustración 14 se puede observar una captura de pantalla del parche digital obtenido luego de aprobar la evaluación.

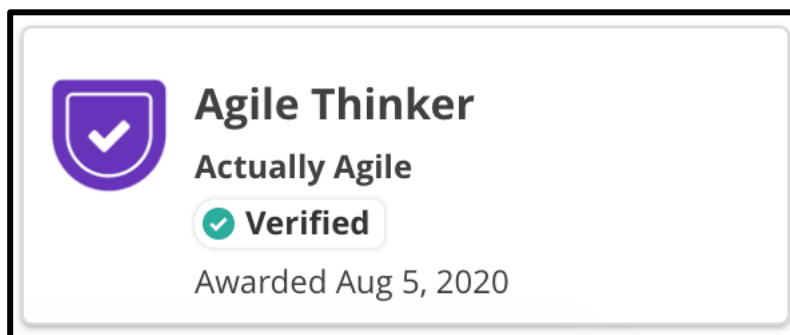


Ilustración 14 – Evidencia de obtención de parche “Agile Thinker”

Fuente: (*Agile Thinker*, 2019)

4.1.3.2 Evaluación para parche “Active Listener Level 1”

La evaluación de este parche consistió en comprobar que se entiende lo que significa ser un oyente activo. Primero se requirió pedirle a dos compañeros de trabajo en puestos de liderazgo y dos compañeros de gremio que mantuvieran un seguimiento del autor por un periodo de 2 semanas para determinar si cumplía con una serie de comportamientos que demostraran que es un oyente activo durante las reuniones en las que estuvieran presentes. Luego se les pidió a estos mismos compañeros que redactaran una especie de reseña resumiendo los comportamientos que observaron durante las 2 semanas. Finalmente, se agendó una reunión para ser evaluado por el CTO de la empresa en base a estas reseñas y a un material de estudio previamente entregado. La evaluación fue realizada por Byron Sommerdahl y fue completada exitosamente.

En la Ilustración 15 se puede observar una captura de pantalla del parche digital obtenido luego de aprobar la evaluación.

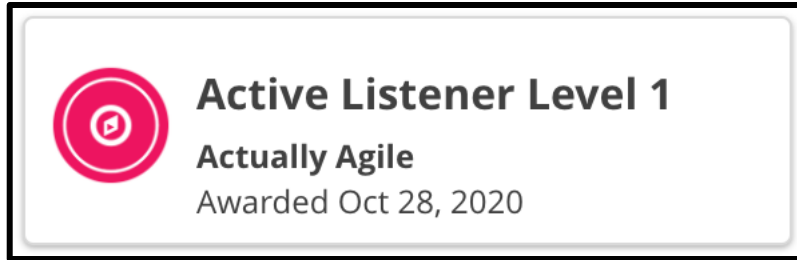


Ilustración 15 – Evidencia de obtención de parche “Active Listener Level 1”

Fuente: (*Active Listener Level 1*, 2020)

4.1.3.3 Clase “English Skill Assessment”

Esta clase retomó el formato de evaluación para parches digitales que se siguió en el entrenamiento al inicio de la práctica y consistió en una serie de evaluaciones que cubrieron diferentes áreas del inglés.

Primero se realizó una prueba de escritura que evaluó el conocimiento de conceptos gramaticales con un formulario de preguntas de selección múltiple en la que se obtuvo una nota de 72%. Luego se agendó una reunión con el emisor del parche correspondiente a esta clase (“English Communicator”) que consistió en una evaluación auditiva y verbal mediante un par de actividades que requirieron que se hablara en inglés por un lapso de tiempo predefinido. Finalmente se desarrolló una tarea que consistió en crear una grabación para evaluar las habilidades de lectura del autor.

Todas las evaluaciones fueron completadas y el parche fue otorgado por el Coordinador de Carreras, Cristian García.

4.1.4 PROYECTOS: MATRIX, OIW Y ACERA

En esta sección se detallan las cuatro actividades más importantes en las que se realizó trabajo (o se estuvo directamente involucrado) para la empresa. Dichas actividades varían en cuanto a su frecuencia y tiempo dedicado, sin embargo, todas recaen en uno de tres proyectos que fueron asignados al autor: Matrix, un proyecto

para un cliente externo, Other Interesting Work (OIW) y Acera, dos proyectos internos.

Matrix es un sitio web dedicado a vender integraciones a compañías de salud digitales, tales como oficinas médicas, clínicas y centros especializados de salud. Por otro lado, OIW es una colección de algunos de los proyectos internos que la empresa está desarrollando actualmente. Este conjunto incluye a Glee2, un sitio web que se planea utilizar como plantilla para futuros proyectos y Recruitment (Reclutamiento), una página web construida para asistir al departamento de Recursos Humanos al momento de documentar candidatos para posteriormente manejarlos según su tipo de contratación (empleados o practicantes). Finalmente, Acera es un sitio web que busca replicar las funcionalidades que provee Badgr (ver marco teórico) con el fin de ahorrarle dinero a la empresa e incorporar dichas funcionalidades con un mayor número de emisores de parches digitales y convertir el sitio en una especie de vitrina de logros para sus empleados.

4.1.4.1 Reuniones diarias

De lunes a viernes, en horarios previamente establecidos, ambos equipos de trabajo de los tres proyectos a los que fue asignado el autor, se reúnen para brindar una actualización sobre el trabajo realizado el día anterior, así como del trabajo que se realizará durante el día, y se observan las tareas pendientes en el Trabajo en Progreso del muro de Trello de los proyectos.

Desde que se invitó a estas reuniones, se ha asistido a todas y cada una de ellas, en ocasiones teniendo que ser el autor quien brinde una actualización en representación del equipo de QA. Incluso se ha dado autorización para crear nuevas tareas en base a descubrimientos realizados en sesiones de prueba.

En un inicio se dificultó vocalizar eficientemente las actualizaciones, pero a medida que se asistió a más reuniones y teniendo en mente la posibilidad de ser el responsable de realizarlas, el autor se prepara todos los días para estar listo en caso de que suceda. Ahora se sigue el estándar de presentación al pie de la letra y se mantiene un claro conocimiento del estado de los proyectos asignados.

En la Ilustración 16 se puede observar una captura de pantalla de una tarea creada para el Trabajo en Progreso del proyecto OIW.

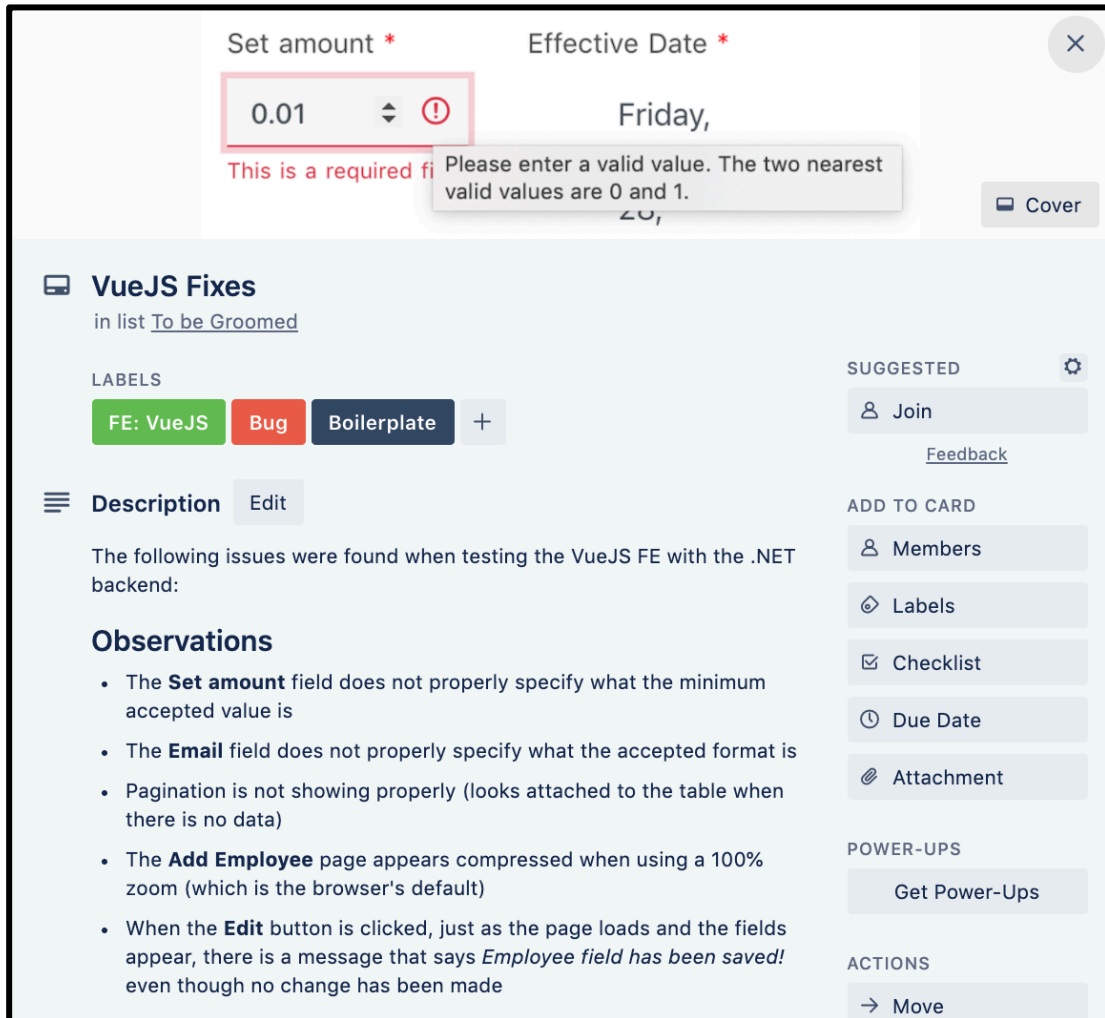


Ilustración 16 – Tarea creada para el Trabajo en Progreso del proyecto OIW

Fuente: (Elaboración Propia)

4.1.4.2 Sesiones de prueba

El trabajo de un Ingeniero en Control de Calidad va muy de la mano con el trabajo de un Desarrollador de Software, por lo que en muchas ocasiones se requiere la colaboración directa de un Desarrollador para llevar a cabo las pruebas. En otros casos (los más comunes), el equipo de QA se reúne para realizar las pruebas

siguiendo una técnica similar a la de Programación en Pareja, donde una persona ejecuta las pruebas mientras el resto supervisa y ofrece alternativas o sugiere correcciones. Por último, están los casos en los que un único Ingeniero en QA realiza las pruebas, sin embargo siempre consulta sus descubrimientos con el Desarrollador responsable y sus compañeros de equipo para corroborar que no se le escape nada.

Se ha participado en todos los escenarios previamente mencionados en su mayoría siendo sesiones de prueba por cuenta propia, especialmente para los proyectos OIW y Acera. En estas ocasiones se han seguido los consejos y las enseñanzas de compañeros más experimentados y se ha puesto en práctica lo aprendido en las sesiones de prueba con el equipo de QA.

La naturaleza de la mayoría de las pruebas ha sido de carácter funcional, pero también se han realizado pruebas de regresión, pruebas exploratorias y pruebas de GUI, apoyándose en los navegadores de búsqueda y demás herramientas pertinentes al proyecto.

Luego de recopilar el tipo de prueba, duración, proyecto asignado y cantidad de errores encontrados durante las 22 semanas que se colaboró a través de 4 proyectos distintos, se brinda el siguiente análisis para cada uno de estos puntos, acompañados de estadísticas.

Como se mencionó anteriormente, el mayor tipo de pruebas que se realizaron fueron pruebas funcionales. Una de estas pruebas típicamente consiste en 3 partes: preparar, actuar y evaluar. *Preparar* consiste en revisar los requerimientos, tanto funcionales como de negocio, y consiste en crear y ordenar la información de prueba que se utilizará. *Actuar* hace alusión a la tarea de revisar dichos requerimientos, ya sea de forma manual o automatizada. Durante la práctica únicamente se realizaron pruebas manuales. Y *evaluar* se dedica a comparar los resultados esperados con los resultados obtenidos y decidir, en base al cumplimiento de los requerimientos, si la prueba es exitosa o no.

Según el análisis completado, se realizaron un total de 58 sesiones de prueba, de las cuales un 74% equivalen a pruebas funcionales. En la Ilustración 17 se puede observar un desglose de los porcentajes para todos los tipos de pruebas realizadas.

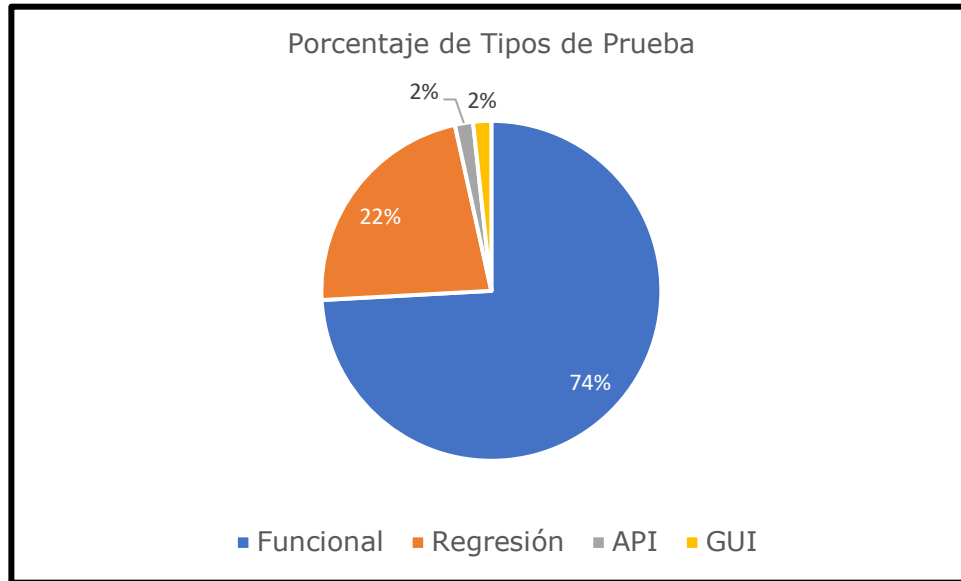


Ilustración 17 - Porcentaje de Tipos de Prueba

Fuente: (Elaboración Propia)

En la Ilustración 18 se puede observar una captura de pantalla de una prueba de regresión realizada para el proyecto OIW.

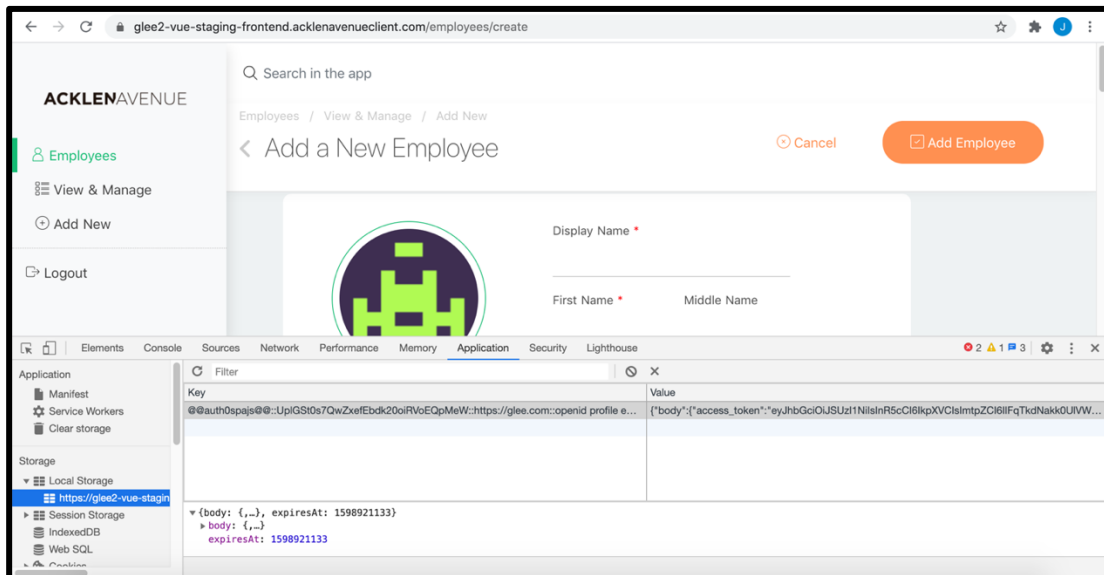


Ilustración 18 – Prueba de regresión realizada para el proyecto OIW

Fuente: (Glee2, 2020)

La duración de una prueba es susceptible a varios factores, tales como el tipo de prueba, la cantidad de casos y escenarios a evaluar, la complejidad de la

información de prueba a crear, entre otros. Así como hay pruebas que pueden ser rápidas y no durar más de media hora, hay pruebas que pueden extenderse por unos cuantos días y que necesiten de varias sesiones para ser completadas.

Según el análisis realizado, en promedio, las sesiones de prueba duraron aproximadamente 61.31 minutos. En la Ilustración 19 se puede observar un desglose de los porcentajes para el tiempo dedicado en cada uno de los proyectos asignados.

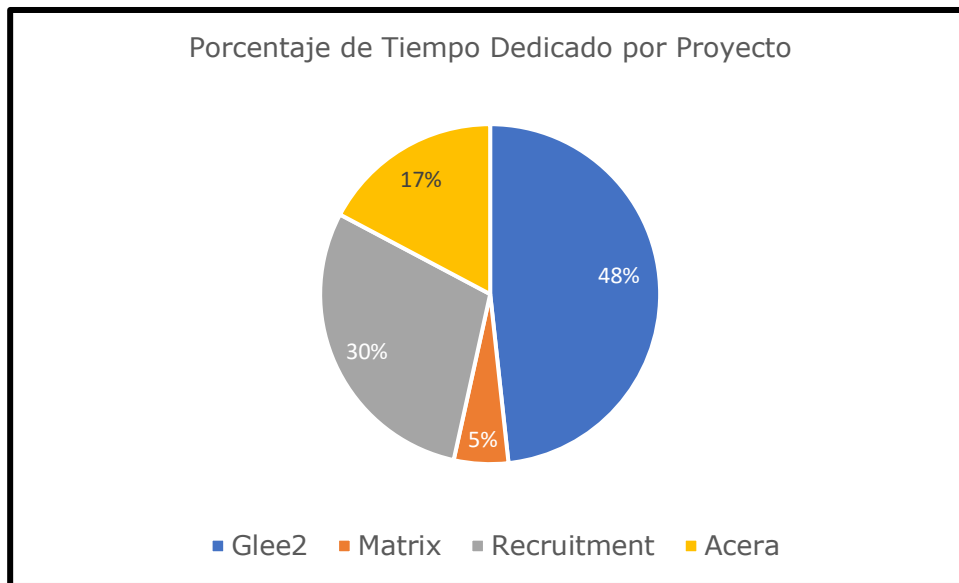


Ilustración 19 - Porcentaje de Tiempo Dedicado por Proyecto

Fuente: (Elaboración Propia)

Es de esperarse que, al empezar como practicante, en un inicio la carga de trabajo no sea la misma que durante las últimas semanas. Este fue el caso para la carga de trabajo asignada en los 4 proyectos en los que se colaboró, en un inicio siendo leve para el proyecto Matrix, ya que contaba con otros dos Ingenieros en QA, pero luego volviéndose pesada al ser el único Ingeniero en QA dedicado para el proyecto Acera. Es importante mencionar que siempre se procuró pasar ocupado mientras no se realizaban pruebas, como se puede apreciar en el Desarrollo Profesional de la sección 4.1.3 y en las vastas reuniones que fueron parte íntegra de la experiencia obtenida.

Según el análisis desarrollado, el proyecto que mayor carga presentó fue Glee2, con un total de 28 sesiones de prueba. En la Ilustración 20 se puede observar un desglose de la carga de trabajo (número de sesiones) asignada en cada uno de los proyectos.

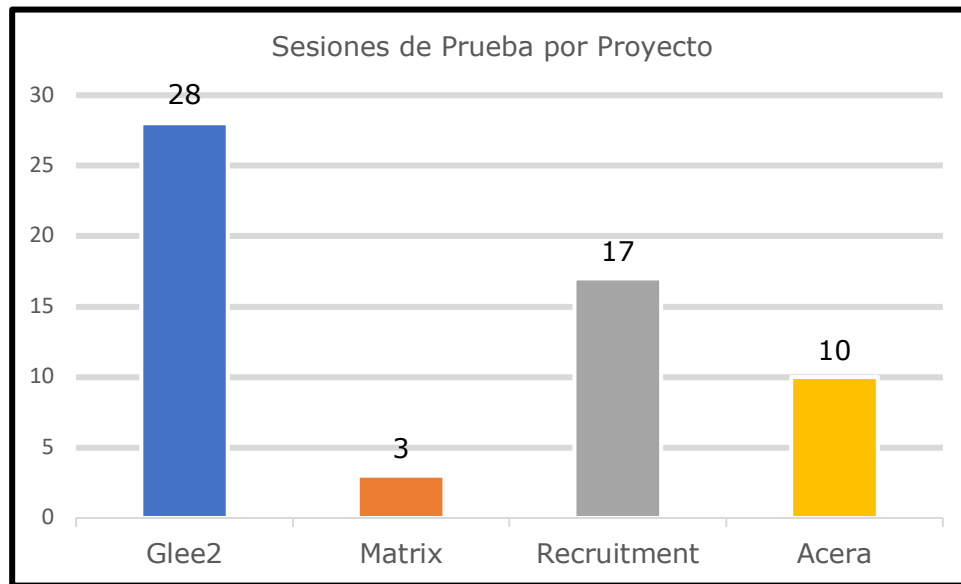


Ilustración 20 - Sesiones de Prueba por Proyecto

Fuente: (Elaboración Propia)

Finalmente, y quizás como punto más importante, se hablará sobre los errores encontrados en las sesiones de prueba.

Antes de adentrarse a este tema, es importante educar al lector con respecto a los dos tipos de errores que pueden encontrarse y a sus diferentes clasificaciones.

En primera instancia, existe un tipo de error llamado *rework*. AA y HU llaman así a los errores que se encuentran en el ambiente de desarrollo o de staging. Por staging, entiéndase un ambiente de pre-producción. El detalle aquí está en que estos errores son descubrimientos que se han hecho antes de que una cierta funcionalidad pase a las manos del cliente. Una vez en sus manos, es decir, en el ambiente de producción, si se llega a encontrar un error, este ahora se considera de tipo *bug*. Por lo tanto, el objetivo principal de los Ingenieros en Control de Calidad es procurar que llegue la menor cantidad de bugs a manos del cliente, por lo que se siguen distintos procesos de documentación dependiendo del tipo de

error encontrado. Para el caso de los proyectos asignados, todos los errores documentados fueron reworks ya que no se realizaron pruebas en producción debido a que 3 de los 4 proyectos fueron internos.

Una vez encontrando un bug o rework y siguiendo la documentación establecida, estos se clasifican según el contexto y el nivel de conocimiento tanto del cliente como del equipo de desarrollo. Según AA, los errores pueden ser clasificados con uno de estos 7 nombres:

- Incumplimiento: el cliente claramente comunicó el requerimiento, el equipo lo entendió claramente y lo documentó, pero su implementación no fue entregada como se especificó.
- No capturado: el cliente claramente comunicó el requerimiento, pero el equipo no comprendió la importancia o no lo capturó con exactitud.
- Desconocido: el cliente no supo cómo o no comunicó el requerimiento y el equipo no pudo haber inferido o hecho una pregunta sobre el mismo.
- Desapercibido: el cliente no supo cómo o no comunicó el requerimiento, pero es algo que el equipo pudo haber inferido razonablemente o expuesto con una pregunta.
- Mala asunción: el cliente no supo cómo o no comunicó el requerimiento y el equipo infirió y documentó un requerimiento que no fue verificado con el mismo; este luego resultó siendo incorrecto.
- Cosmético: el cliente requiere ajustes que no afectan la funcionalidad del sistema pero son necesarios para satisfacer los requerimientos de aspecto y tacto (UI/UX).
- Cambio: el cliente claramente comunicó el requerimiento, el equipo lo entendió claramente y lo documentó y la implementación entregada se apegó a la especificación, pero luego que el cliente la revisó, decidió hacerle un cambio al requerimiento.

Según el análisis completado, se encontraron un total de 16 reworks. En la Ilustración 21 se puede observar un desglose de los porcentajes de las clasificaciones que se le dieron a los errores encontrados.

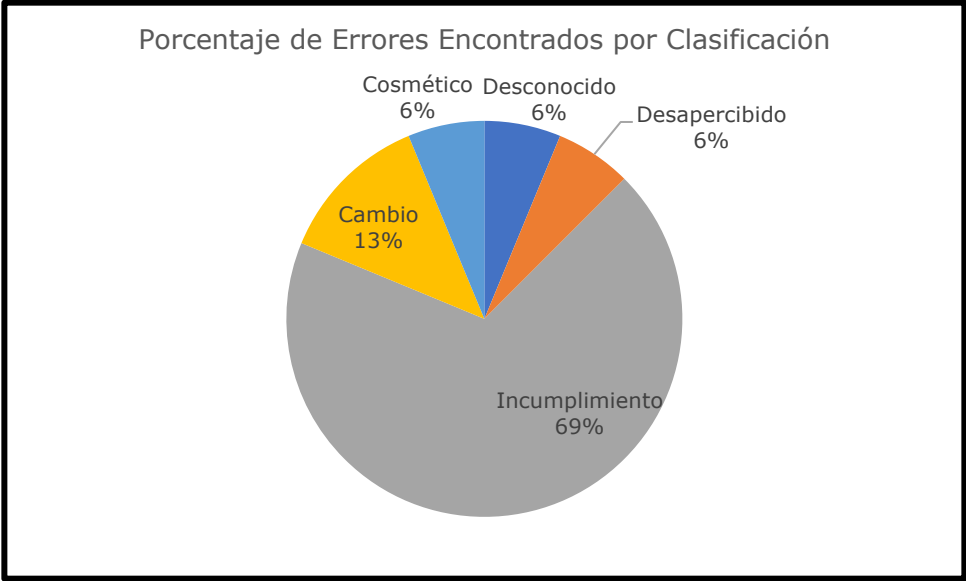


Ilustración 21 - Porcentaje de Errores Encontrados por Clasificación

Fuente: (Elaboración Propia)

En la Ilustración 22 se puede observar una captura de pantalla de la documentación realizada para un error encontrado en el proyecto Acera.

LABELS

Overlooked +

Description Edit

If a user was awarded a badge that later was manually removed from Badgr via the *Remove from Backpack* feature, they should be able to claim it again through Acera.

Steps to Reproduce

1. Remove a badge from your backpack
 - Go to <https://info.badgr.com>
 - Click **Sign In** on the top right corner
 - Sign in to Badgr with an account that has badges that are also in Acera*
 - Click **Backpack** on the top right corner
 - Click on the name of a badge that is also in Acera
 - Click **Remove from Backpack** on the right hand side
 - Click **Remove Badge**
2. Attempt to reclaim the badge
 - Go to <https://acera-dev.web.app/login>
 - Log in to Acera
 - Log in to Badgr's OAuth with the same account you removed the badge from
 - Click on **Pathways** on the top left corner
 - Find a pathway that contains the badge you just removed from your backpack
 - Click **View**
 - Click on the badge's node
 - Click **Claim badge** on the right hand side
 - Input your email and anything else on the evidence field

*Remember that as of now, only Acklen Avenue, Actually Agile, and CodeX Academy are able to award badges through Acera.

Actual Result

User is unable to reclaim the badge and the following message shows up:
"You have already claimed this badge!".

SUGGESTED

Join

ADD TO CARD

Members

Labels

Checklist

Due Date

Attachment

Cover

POWER-UPS

Card Priority

+ Add Power-Ups

BUTLER NEW ⓘ

+ Add Card Button

ACTIONS

→ Move

Copy

Make Template

Watch

Archive

Ilustración 22 - Ejemplo de Documentación de Errores en Proyecto Acera

Fuente: (Elaboración Propia)

4.1.4.3 Reuniones semanales de preparación (pre-grooming)

Cada semana, para el proyecto Matrix, el equipo se reúne para preparar las tareas que se añadirán al Trabajo en Progreso con el objetivo de elaborar preguntas para el cliente que serán contestadas en una reunión semanal aparte.

Se ha participado como observador en estas reuniones, que han ayudado a entender más a detalle el proceso de desarrollo que la empresa sigue para sus proyectos.

4.1.4.4 Reuniones semanales con el cliente (grooming)

Luego de haber realizado la reunión de preparación con el equipo, un día después este se vuelve a reunir y se lleva a cabo la reunión semanal con el cliente, en la que se discuten las nuevas tareas a ser elaboradas y se aclaran todas las dudas posibles para aproximarse lo más cercano a sus expectativas.

Se ha participado como observador en estas reuniones y han sido de mucho beneficio para adquirir experiencia de primera mano en lo que se refiere a construir una buena relación con los clientes y la importancia de mantener una constante interacción con ellos.

4.1.5 ACKLEN AVENUE Y HERO UNIT

Las siguientes actividades no pertenecen a ningún proyecto en particular, tampoco son de ningún gremio en específico, pero sí integran a todos y cada uno de los empleados de la empresa y por ende, al autor. Estas han sido categorizadas como actividades organizacionales y se detallan a continuación.

4.1.5.1 Reuniones mensuales (All Hands)

Cada última semana del mes se reúne todo el personal de la empresa, desde el Director Ejecutivo hasta el practicante más nuevo, y cada tribu brinda una actualización de los proyectos en los que están trabajando. Se comparten actualizaciones sobre las finanzas, ventas, reclutamiento y mercadeo, y se discuten iniciativas estratégicas. Finalmente, se dedica un espacio para que todos puedan hacer preguntas y se presentan a los nuevos Ackleners y Heroes.

En su primera All Hands, el autor se presentó y compartió un dato curioso sobre él, haciendo su primera aparición oficial frente a toda la empresa como uno de los nuevos Ingenieros en Control de Calidad.

En la Ilustración 23 se puede observar una captura de pantalla del momento en el que el autor se presentó durante la reunión mensual.

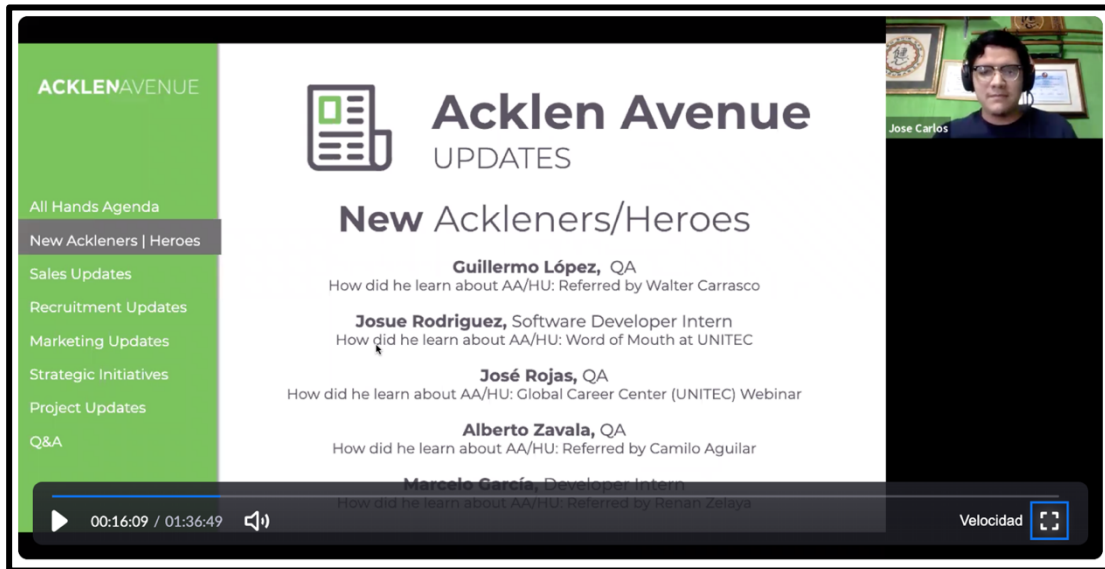


Ilustración 23 – Evidencia de participación en reunión mensual “All Hands”

Fuente: (Elaboración Propia)

4.1.6 PUESTO COMO CONSULTOR EN PICA

Finalmente, más o menos un mes y medio antes de concluir la práctica, se presentó la oportunidad de optar por un puesto de trabajo remunerado a través de Hero Unit como consultor de QA en una empresa llamada PICA.

Antes de profundizar en los detalles del puesto, se considera importante mencionar que este será remunerado una vez que la práctica profesional con Acklen Avenue y Hero Unit termine. Es decir, no se recibió ningún tipo de compensación monetaria desde que se empezó en el puesto hasta después de haber terminado la práctica. Asimismo, cabe destacar que en esta sección únicamente se estará brindando un panorama general del trabajo realizado hasta la fecha en vista de dos razones. La primera siendo que recién se viene empezando a incorporar a los grupos de trabajo de la empresa y el alcance del puesto aún no está completamente definido, y la segunda siendo que la empresa es bastante cuidadosa con la seguridad de su información, a tal punto que se requirió firmar un acuerdo de confidencialidad en el que se prohíbe compartir muchos detalles del trabajo realizado y los procesos involucrados.

PICA (ProAssurance Insurance Company of America) es una empresa subsidiaria de ProAssurance que se dedica a proveer seguros de negligencia médica a podólogos, quiroprácticos y odontólogos. El puesto para el que se fue contratado tiene como objetivo compartir la experiencia que se tiene en control de calidad para ayudar a sus equipos de trabajo a mejorar sus procesos de pruebas implementando buenas prácticas y estándares de la industria. También se planean introducir herramientas que apoyen a estos procesos y herramientas especializadas para técnicas particulares de prueba, como pruebas de regresión automatizadas o pruebas funcionales de APIs. Hasta el momento, el trabajo ha consistido principalmente en realizar investigaciones que ayuden a descubrir herramientas y recursos que faciliten el cumplimiento de los objetivos del puesto, ya sean completamente nuevas o que se hayan utilizado previamente durante la práctica o de manera consistente en AA y/o HU. También se han realizado investigaciones para indagar en la industria de seguros de negligencia médica en vista que se necesita un amplio conocimiento sobre los conceptos y pólizas para los proyectos en los que se colabore. Por último, se han realizado unas cuantas pruebas funcionales y de APIs para un proyecto interno que maneja un portal mediante el cual se gestiona el sistema de pólizas de seguros de negligencia médica de los clientes de la empresa.

4.2 CRONOGRAMA DE ACTIVIDADES

En la Ilustración 24 se puede observar el Cronograma de Actividades como resultado de las diferentes tareas realizadas durante la práctica profesional.

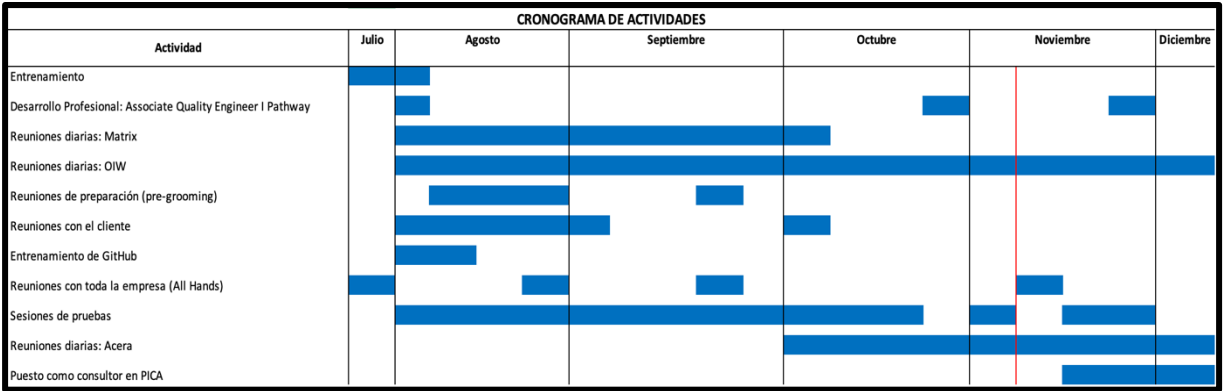


Ilustración 24 – Diagrama de Gantt

Fuente: (Elaboración Propia)

V. CONCLUSIONES

- Se logró asistir en un total de 4 proyectos distintos (Matrix, Glee2, Recruitment, Acera) en los cuales, siguiendo las prácticas de desarrollo ágil y colaborativo aplicadas en la empresa, se brindaron soluciones de software en respuesta a los errores encontrados en más de 50 sesiones de pruebas que abarcaron temas de seguridad, experiencia de usuario, diseño, entre más, logrando así garantizar un estándar de calidad elevado para dichos proyectos y la empresa como tal.
- Siguiendo un método similar al de programación en pareja, similar en el sentido que se trabajó junto a desarrolladores pero no se compartieron tareas de programación, es decir, únicamente se cumplió con el rol de supervisar el código y ofrecer sugerencias, se propusieron nuevas funcionalidades en base a los errores y descubrimientos que surgieron de los proyectos cuando se realizaron sesiones de prueba junto a estos desarrolladores. La implementación de estas sugerencias quedó a discreción de los mismos.
- A pesar de que la mayor parte del tiempo no se compartió el código de los proyectos con el autor, en algunas ocasiones y para casos muy específicos sí se logró visualizarlo junto al desarrollador que lo elaboró. En estos momentos se mencionaron todas las buenas prácticas y posibles mejoras al código que se consideraron apropiadas para el caso, las cuales también quedaron a discreción de los desarrolladores.
- Se participó en más de 50 sesiones de prueba, al inicio junto al equipo de QA pero luego, y en su mayoría, por cuenta propia, en las cuales se realizaron pruebas de regresión, funcionales, de GUI, en pareja, exploratorias, de integración, entre más. Asimismo, se siguieron todos los lineamientos de la empresa para documentar y reportar errores, logrando así velar por el correcto funcionamiento de los proyectos asignados.

VI. RECOMENDACIONES

1. Se debe incluir más a menudo la retroalimentación de los TTLs y los diseñadores de UI/UX en las sugerencias brindadas por los Ingenieros en Control de Calidad con el fin de proveer un punto de vista técnico e imparcial, siempre enfocado en mejorar la calidad de los proyectos.
2. Se debe compartir el código de los proyectos con los Ingenieros en Control de Calidad, no porque estos vayan a programar los cambios ellos mismos, pero porque para aquellos con antecedentes computacionales y experiencia en programación, puede resultar más fácil identificar la causa de un error, entenderlo y explicárselo de una mejor manera a los desarrolladores.
3. Se debe procurar contar con un ambiente de pruebas para todos los proyectos que involucren Ingenieros en Control de Calidad. No tener este ambiente no solo no garantiza tener el mismo estándar de calidad en todos los proyectos, sino que también puede provocar que el flujo de trabajo se estanque cuando no se pueda cargar un cambio a un ambiente en los casos que el ambiente de pruebas se comparta con el ambiente de desarrollo.
4. Se debe procurar designar las responsabilidades que normalmente tendría un cliente en un proyecto interno a algún Product Owner, Scrum Master o TTL que disponga de más tiempo cuando el cliente claramente no disponga del tiempo necesario para cumplir con sus tareas. Esto con el objetivo de acelerar el flujo de trabajo cuando una única acción del cliente provoque que todo el equipo deba esperar a que este la cumpla para que el proyecto pueda seguir en marcha.

BIBLIOGRAFÍA

1. *12 Principles Behind the Agile Manifesto*. (2015). Recuperado 13 de septiembre de 2020, de <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>
2. *15-1132.00—Software Developers, Applications*. (s. f.). Recuperado 28 de agosto de 2020, de <https://www.onetonline.org/link/summary/15-1132.00>
3. *About Zoom*. (s. f.). Zoom Video. Recuperado 12 de septiembre de 2020, de <https://zoom.us/>
4. *Adding lists to a board*. (2019). Recuperado 12 de septiembre de 2020, de <https://help.trello.com/article/744-adding-lists-to-a-board>
5. *Agile Thinker*. (2019). Recuperado 28 de agosto de 2020, de <https://badgr.com/pathways/5dc078b146e0fb0022c8f6a5:email:jose.carlos@herounit.io>
6. *Atmosphere of Creativity*. (2019). Recuperado 28 de agosto de 2020, de <https://badgr.com/pathways/5deefcf46e0fb00214099d6:email:jose.carlos@herounit.io>
7. *Back-End Web Architecture*. (s. f.). Codecademy. Recuperado 28 de agosto de 2020, de <https://www.codecademy.com/articles/back-end-architecture>
8. Blackstock, S. (2018). *Customer Service vs Customer Experience: Explained - Help Scout*. Recuperado 14 de septiembre de 2020, de <https://www.helpscout.com/blog/customer-service-vs-customer-experience/>
9. Crispin, L., & Gregory, J. (2009). *Agile testing: A practical guide for testers and agile teams*. Addison-Wesley.

10. Dalto, J. (2020). A3: What Is the A3 Problem-Solving Method? *Convergence Training Blog*. Recuperado 14 de septiembre de 2020, de <https://www.convergencetraining.com/blog/a3-a3-problem-solving-method>
11. *DevOps Engineer*. (s. f.). Recuperado 28 de agosto de 2020, de <https://acklenavenue.com/careers/devops-engineer>
12. DiMaria, F. (2016). *5 Quick Tips on How To Use Google Classroom*. Recuperado 29 de agosto de 2020, de <https://www.aeseducation.com/blog/2016/10/how-to-use-google-classroom>
13. Freeman, J. (2019). *What is an API? Application programming interfaces explained*. InfoWorld. Recuperado 15 de septiembre de 2020, de <https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html>
14. Garcia, C., & Sommardahl, B. (2019a). *Community Champion Badge*. Recuperado 28 de agosto de 2020, de <https://classroom.google.com/u/3/w/Mzg5NjIzMzMzMjda/t/all>
15. Garcia, C., & Sommardahl, B. (2019b). *Culture Champion Badge*. Recuperado 28 de agosto de 2020, de <https://classroom.google.com/u/3/w/NDE5ODM4NDcwNDVa/t/all>
16. Garcia, C., & Sommardahl, B. (2019c). *Team Oriented Badge*. Recuperado 28 de agosto de 2020, de <https://classroom.google.com/u/3/w/MzkxMDI5MDQyNTFa/t/all>
17. Garcia, C., Sommardahl, B., & Rocco, T. (2019a). *Our Toolbox*. Recuperado 28 de agosto de 2020, de <https://classroom.google.com/u/3/w/NDE5ODM3NjI1NDRa/t/all>

18. Garcia, C., Sommardahl, B., & Rocco, T. (2019b). *Our Vision and Values*. Recuperado 28 de agosto de 2020, de <https://classroom.google.com/u/3/w/Mzg5NjIzMDg3OTNa/t/all>
19. Ghahrai, A. (2019). *Types of Software Testing*. Recuperado 15 de septiembre de 2020, de <https://devqa.io/types-of-testing/>
20. Gilbert, N. (2020). *Jibble Review*. FinancesOnline. Recuperado 12 de septiembre de 2020, de <https://reviews.financesonline.com/p/jibble/>
21. *Hello World – GitHub Guides*. (2020). Recuperado 12 de septiembre de 2020, de <https://guides.github.com/activities/hello-world/>
22. Iafrate, M. (2017). *Digital Badges: What Are They And How Are They Used?* ELearning Industry. Recuperado 29 de agosto de 2020, de <https://elearningindustry.com/guide-to-digital-badges-how-used>
23. Javier. (2017). Gestión de proyectos con Post it. *Blog de 20milproductos*. Recuperado 13 de septiembre de 2020, de <https://www.20milproductos.com/blog/gestion-de-proyectos-con-post-it/>
24. John, S. (2020). «*What is Google Meet?*»: A guide to Google's video-chatting service, including pricing options and how to join a meeting. Business Insider. Recuperado 12 de septiembre de 2020, de <https://www.businessinsider.com/what-is-google-meet>
25. Karch, M. (2019). *What is Google Calendar?* Lifewire. Recuperado 12 de septiembre de 2020, de <https://www.lifewire.com/google-calendar-1616582>
26. Meltzer, D. (2018). *Go the Extra Mile If You Want to Achieve Excellence*. Entrepreneur. Recuperado 14 de septiembre de 2020, de <https://www.entrepreneur.com/article/311524>

27. Mezquita, F. (2018). *What defines us—Acklen Avenue, a software development company*. Recuperado 29 de agosto de 2020, de <http://acklenavenue.com/blog/defining-acklen-avenue>
28. Navas, M. A. (2018). *¿Qué es la ley de Moore y para qué sirve?* Professional Review. Recuperado 29 de agosto de 2020, de <https://www.profesionalreview.com/2018/04/01/que-es-la-ley-de-moore-y-para-que-sirve/>
29. Novabench. (s. f.). Recuperado 12 de septiembre de 2020, de <https://novabench.com/>
30. *Pair Programming*. (s. f.). IONOS Digitalguide. Recuperado 28 de agosto de 2020, de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/pair-programming/>
31. Progressa. (2014). *¿Qué es la Mejora Continua?* Progressa Lean. Recuperado 13 de septiembre de 2020, de <https://www.progressalean.com/que-es-la-mejora-continua/>
32. Qué es SCRUM. (2008). *Proyectos Ágiles*. Recuperado 28 de agosto de 2020, de <https://proyectosagiles.org/que-es-scrum/>
33. Reyes, G. (s. f.). *They Say, "Be Agile". What Does that Really Mean?* The Daily Mindflash. Recuperado 13 de septiembre de 2020, de <https://mindflash.com/blog/they-say-be-agile-what-does-that-really-mean>
34. Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular agile process*. Addison-Wesley.
35. Sommardahl, B. (2019). *Pathway Execution Sequence*. Recuperado 28 de agosto de 2020, de <https://youtu.be/WwPfZ0jrbiI>

36. Somnardahl, B., Garcia, C., & Gonzales, Y. (2019). *Dojo Orientation Quiz*. Recuperado 28 de agosto de 2020, de https://docs.google.com/forms/d/e/1FAIpQLSfZ65p9C6rK3iFQ2F8gu93dZUFJpbnOThr_Z8CmhUQaqfmZng/viewscore?hr_submission=ChgIh420mYIEEg8I7OPk0JEBEGYI_YS09VkQAAQ&viewscore=AE0zAgACeBaQPSR73lpIYt_vqSa2F0e5iEfXlqF50onwP0GY0zODh-8gHms
37. Tena, M. (2020). Metodología «Agile». La revolución de las formas de trabajo. *BBVA NOTICIAS*. Recuperado 13 de septiembre de 2020, de <https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/>
38. Wester, J. (2012). What is Kanban? *Everyday Kanban*. Recuperado 12 de septiembre de 2020, de <http://www.everydaykanban.com/what-is-kanban/>
39. *What Is a Front-End Developer?* (s. f.). Recuperado 28 de agosto de 2020, de <https://frontendmasters.com/books/front-end-handbook/2018/what-is-a-FD.html>
40. *What is GitLab?* (s. f.). GitLab. Recuperado 14 de septiembre de 2020, de <https://about.gitlab.com/what-is-gitlab/>
41. *What is Gmail*. (s. f.). Recuperado 12 de septiembre de 2020, de <https://www.javatpoint.com/gmail>
42. *What is Slack?* (s. f.). Slack Help Center. Recuperado 12 de septiembre de 2020, de <https://slack.com/help/articles/115004071768-What-is-Slack->
43. *What is Trello?* (2017). Recuperado 12 de septiembre de 2020, de <https://help.trello.com/article/708-what-is-trello>

44. *Active Listener Level 1*. (2020). Recuperado 7 de noviembre de 2020, de https://badgr.com/public/assertions/d6CFG4GIQvqoz0_tguOosw
45. Garcia, C. (2020). *English Writing Assessment*. Recuperado 2 de diciembre de 2020, de https://docs.google.com/forms/d/e/1FAIpQLSef7LoXUxyzLbLfjNeXWH36WixrLYvHgU4fCHEHUT-m82amYw/viewscore?viewscore=AE0zAgD18aDku3kJ-yDe97_qoSRRGFXESUj7d9CFYnIT&gxids=7628