

UNIVERSIDAD DE SONORA

DIVISIÓN DE INGENIERÍA

Departamento de Ingeniería Industrial



Nearsoft Academy

Reporte de prácticas profesionales

Presenta:

Cristian Ariel Cota Gamboa

INGENIERO EN SISTEMAS DE INFORMACIÓN

Director: Dr. Alonso Perez Soltero

ÍNDICE GENERAL

1. INTRODUCCIÓN	3
1.1 BREVE EXPLICACIÓN DEL PROYECTO	5
1.2 OBJETIVOS DEL PROYECTO	6
2. DESCRIPCIÓN DEL CONTEXTO	8
2.1 EQUIPAMIENTO E INSTALACIONES	10
2.1 CLIENTES DE LA EMPRESA	11
3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS	13
4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS	18
4.1 FASE 1. RESET	18
4.2 FASE 2. CONTRIBUIR AL CÓDIGO ABIERTO	22
4.3 FASE 3. CREAR UN PROYECTO DESDE CERO	24
5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA	27
5.1 ANÁLISIS GENERAL DEL PROYECTO	27
5.2 ANÁLISIS DE LOS OBJETIVOS DE LAS PRÁCTICAS	27
5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS	28
6. CONCLUSIONES Y RECOMENDACIONES	29
6.1 RECOMENDACIONES AL PROGRAMA DE ISI	30
7. REFERENCIAS	31

ÍNDICE DE FIGURAS

FIGURA 1.1 LOGOTIPO DE NEARSOFT ACADEMY	5
FIGURA 2.1 LOGOTIPO DE NEARSOFT	7
FIGURA 2.2 MATT PEREZ (COO) & ROBERTO MARTÍNEZ (CEO)	9
FIGURA 2.3 EDIFICIO DE NEARSOFT HERMOSILLO	10
FIGURA 2.4 PRINCIPALES CLIENTES DE NEARSOFT	12
FIGURA 4.1 PORTADA DE THE PRAGMATIC PROGRAMMER	19
FIGURA 4.2 ARQUITECTURA DEL SISTEMA RECOMENDADOR CON EL USO DE MAHOUT	20
FIGURA 4.3 PRETOTIPO DE IDENTIFICACIÓN DE ESPACIOS EN ESTACIONAMIENTO	21
FIGURA 4.4 PÁGINA INICIAL DEL SISTEMA TRAINING	22
FIGURA 4.5 DISEÑO ORIGINAL DE LA VENTANA	24
FIGURA 4.6 PRUEBAS CON RASPBERRY PI Y GOPRO	25
FIGURA 4.7 PORTAL FINALIZADO	25
FIGURA 4.8 PORTAL FUNCIONANDO CON PERSPECTIVA DE LA OFICINA DE HERMOSILLO	26

1. INTRODUCCIÓN

Como parte de la conclusión de nuestros estudios superiores, la Universidad de Sonora nos pide acercarnos a una empresa con el giro a nuestra carrera y pedir que nos acepten para que durante 340 horas nos acepten como sus practicantes y nosotros podamos aprender de las tareas que ahí nos asignen. Personalmente me acerqué a la empresa Nearsoft debido a que conozco la calidad del trabajo que realizan, además de que la cultura de trabajo que se mantiene dentro no es alcanzada por ninguna otra empresa con el mismo giro dentro de la región.

Actualmente las empresas realizan un gran número de sus procesos aprovechando el uso de las tecnologías de información (TI) las cuales si se usan de la forma correcta pueden ser de mucho beneficio, ahorrando tiempo, mejorando la preservación de los datos y su posterior consultas, agilizando procesos claves, liberando espacio físico, menos gasto de materiales de papelería y demás ventajas. Es por eso que todas las empresas, sin importar su clasificación están buscando migrarse al uso de estas tecnologías. Nearsoft ofrece como producto el integrarse al equipo de desarrollo de una empresa para poder acelerar el desarrollo de sus plataformas en construcción y ayudarlas con su crecimiento.

Para que la empresa receptora me aceptara como su practicante, fue necesario pasar por un proceso de reclutamiento por el cual han pasado todos los empleados de la empresa. Primeramente es aplicado un examen de lógica, que básicamente plantea un lenguaje de programación inexistente y conforme pasan las preguntas sube la dificultad de este, todas las preguntas tienen relación a las respuestas anteriores y no debes pasar nada por alto, esto para ver como esta tu capacidad de entender las cosas en un tiempo corto y ver como te manejas ante esas situaciones, el examen dura 30 minutos, es en inglés y en línea. Si acertaste al menos 16 de 20 preguntas, estás listo para la siguiente etapa, que es una entrevista en inglés con el co-fundador de Nearsoft. Esta entrevista no dura más de 10 minutos y solamente se busca que tengas el nivel suficiente para poder comunicarte con los demás. Finalmente, para ser aceptado con éxito es necesaria una pequeña entrevista

realizada por Isaac López, que es el líder el Nearsoft Academy, donde se platica que has realizado anteriormente, si has trabajado, qué tecnologías dominas, entre otras cosas, básicamente es para conocer al candidato y ver el potencial que se tiene. Al concluir esta entrevista Isaac te dice si eres o no aceptado. Gracias al éxito en las 3 etapas mencionadas anteriormente fui aceptado como interno en la generación 2016A del Nearsoft Academy, junto con otros 13 integrantes. De esos 7 éramos de Hermosillo, 5 de la oficina de Chihuahua y 2 de Ciudad Obregón.

En el siguiente documento se relata cuáles fueron la actividades realizadas a lo largo de nuestra estancia dentro de Nearsoft, cuales fueron los retos, objetivos y cómo es que fueron o no cumplidos y cuáles sus motivos.

1.1 BREVE EXPLICACIÓN DEL PROYECTO



Figura 1.1 Logotipo de Nearsoft Academy

Nearsoft Academy (figura 1.1) es un programa que consiste en enlistar un grupo de estudiantes a punto de terminar sus estudios superiores o recientemente egresados en una carrera relacionada a las tecnologías de información, con el objetivo de conseguir un lugar dentro de uno de los clientes con los que cuenta la empresa. Este proyecto está planificado para que el interno consiga un cliente dentro de los 6 meses que tiene como duración. Cada mes se tiene una tarea diferente para todos ellos, a las cuales se le denominan fases.

Durante la primer fase se introduce al interno al funcionamiento de la empresa, cual es su misión y visión y como un nuevo integrante de ella puede aportar lo necesario para seguir con el crecimiento de la misma. Durante este mes se intenta hacer que el interno pierda las malas costumbres con las que uno vive cotidianamente y pueden resultar contraproducentes a la hora de obtener un trabajo real, esto se logra por medio de lecturas, cursos, tareas semanales, ejercicios, entre otras actividades.

El segundo mes consiste en trabajar en un proyecto desde cero, donde todos los miembros del equipo de internos trabajan juntos con un fin en común. Durante este mes se hace uso de metodologías ágiles, el uso de tecnologías nuevas, realización de presupuesto, administración de tareas, etc.

La tarea del interno a lo largo de la tercer fase, es contribuir a la comunidad del código abierto, que consiste básicamente en aportar algunas líneas de código que

ayuden a resolver un problema dentro de un proyecto de software que usa la comunidad de desarrolladores actualmente. No solo es arreglar problemas sino que también se toma en cuenta dentro de la comunidad el hacer ver puntos de vista que uno tiene de las contribuciones dentro de los repositorios de los proyectos.

Los tres meses restantes y última fase es conseguir un cliente dentro de las oportunidades que Nearsoft nos presenta, aquí el interno se prepara en la tecnología que más le llame la atención para poder obtener un puesto en lo que prefiere y se desenvuelva mejor. Durante esta etapa, los miembros se preparan por medio de lecturas, cursos y entrevistas de prueba, las cuales son realizadas por desarrolladores actuales dentro de la misma empresa.

1.2 OBJETIVOS DEL PROYECTO

Durante la prolongación del proyecto, el objetivo principal es que el interno sea lo suficientemente capaz para desenvolverse y ocupar un puesto como desarrollador de software dentro de uno de los clientes con los que Nearsoft cuenta. Para lograr alcanzar este objetivo es necesario que los integrantes identifiquen sus malos hábitos de comunicación y cualquier mal comportamiento que se tenga al trabajar en equipo. Es necesario también aprender a utilizar herramientas que agilicen el trabajo, así como el proceso de aprendizaje. El saber transmitir ideas, no solo a unas cuantas personas sino a un gran público. Es necesario también el aprender a transmitir el conocimiento a los demás y no solo esperar el crecimiento personal, sino el hacer que todos crezcamos como el equipo del que formamos parte.

En la parte técnica es necesario aprender diferentes paradigmas de programación y estar conscientes sobre cuáles pueden ser aplicados en los diferentes contextos que se nos llegan a presentar en situaciones cotidianas, no debemos enfocarnos en crecer solamente en una herramienta o lenguaje específico, ya que para eso existe una gran diversidad de estos. También es requerido el aprender a manejar tecnologías que sirven como manejadores de versiones, que en el caso de Nearsoft es *Git* y nos ayuda con el trabajo en equipo y la colaboración a un proyecto, sin

importar el número de personas con las que está conformada ni la distancia que exista entre los miembros, todos trabajan con el trabajo que todos realizan día a día. También se ponen en acción metodologías de desarrollo de software y cómo es trabajar con un proyecto real antes de intentar cubrir un puesto.

Un objetivo fundamental es ser capaz de escribir código funcional, entendible, escalable, que cumpla con los estándares que pide quien lo requiera y pueda ser probado con facilidad, es por eso que también es necesario conocer las herramientas para realizar pruebas y como son las metodologías usadas dentro de este ambiente.

El estar preparado para una entrevista con un cliente real es de suma importancia al momento de buscar cubrir con una vacante que se encuentre disponible, esto puede ser logrado con mayor seguridad si cumplimos lograr todos los objetivos mencionados anteriormente y para lograr conseguir con esto, también se pone a disposición de los practicantes, las entrevistas que se realizan día con día a los desarrolladores que intentan entrar a Nearsoft y ver qué es lo que se debe saber para poder salir exitoso de una entrevista.

2. DESCRIPCIÓN DEL CONTEXTO



Figura 2.1 Logotipo de Nearsoft

Nearsoft (figura 2.1) es una empresa dedicada al desarrollo de software, que cuenta con un amplio número de clientes con un alto nivel comercial en el mundo. Sus clientes se encuentran en los Estados Unidos de América y su lugar de operaciones se da en México. Nearsoft ha sido premiada durante los últimos años (a partir del 2008) consecutivamente dentro de las 10 mejores empresas para trabajar en México por *Great Place to Work* quien gracias a una serie de encuestas y estudios ha otorgado estas menciones a la empresa. Además también ha sido listado dentro de los lugares más democráticos para trabajar por *World Blu* desde el 2009.

Existe un número de personas que son reconocidas dentro de la empresa, las cuales son líderes en diferentes áreas, como Roberto Martínez (CEO) quien fundó la empresa hace más de 10 años. Este hombre fue egresado del Instituto Tecnológico y de Estudios Superiores de Monterrey en el año de 1998 y fundó Nearsoft sin llevar ese nombre el año de 1999. Años más tarde se asoció con Matt Pérez (COO) quien es graduado de la Universidad de Illinois, en Chicago. Al momento de ocurrir esta asociación se fundó oficialmente Nearsoft en el año 2006. Con apenas un pequeño grupo de personas menor a 15 personas, actualmente hay más de 230 empleados dentro de la empresa. En la figura 2.2 podemos apreciar a estos 2 personajes.



Figura 2.2 Matt Perez (COO) & Roberto Martinez (CTO)

A pesar de las personas resaltadas anteriormente, es necesario resaltar que esta empresa no tiene un organigrama, es decir, es una empresa con estructura horizontal, donde no existen los puestos superiores entre sí. Esto favorece las relaciones entre las personas, el crecimiento general de los empleados de manera profesional y personal.

Nearsoft cuenta con una ideología de trabajar muy diferente a lo que se maneja actualmente a nuestro alrededor dentro de la sociedad en la que fuimos educados, siendo poseedora de una cultura ampliamente conocida por todo aquel que se asome dentro de la empresa, reconocida también por su calidad de trabajo, por la transparencia entre la información que maneja dentro, sin excluir a ningún miembro de ella, convirtiéndose así, en una de las mejores empresas para trabajar dentro país y con reconocimiento mundial.

2.1 EQUIPAMIENTO E INSTALACIONES

La empresa cuenta con 3 oficinas dentro de la república mexicana, dispersas en 3 grandes ciudades: Hermosillo, Chihuahua y Ciudad de México, siendo Hermosillo la que cuenta con la oficina principal y se encuentra ubicada en el Blvd. Antonio Quiroga número 21 en la colonia El Llano. Además cuenta también con una oficina en San José, California.



Figura 2.3 Edificio de Nearsoft Hermosillo

El edificio de la ciudad de Hermosillo (figura 2.3) cuenta con 3 plantas, en la primera se ubica un auditorio para juntas, talleres, exposiciones y cualquier actividad deseable donde se necesite este espacio, también se cuenta con un gimnasio, baños con regadera, cocina y comedores, un área de recepción y las oficinas de finanzas, además de contar con un área para convivencia. La segunda planta cuenta con áreas específicas para trabajar, tanto a desarrolladores, diseñadores gráficos, reclutadores y otras áreas, también se ubica un espacio para el site de la empresa, hay una sala específica para juntas, esta planta además cuenta con un área de juegos y de descanso y en el exterior se ubica una pequeña terraza. La tercera y última planta cuenta con el mismo espacio para trabajar que la segunda,

una segunda sala de juntas igual a la de la planta debajo a ella y también cuenta una terraza con una mayor superficie en el exterior.

Debido a la facilidad de movilidad que se brinda a los empleados es difícil conocer el número de equipos con los que se cuenta, pero es sabido que a cada empleado se le brinda computadora personal para trabajar desde el lugar de su conveniencia. A cada desarrollador se le entrega una MacBook Pro, que sea del modelo más reciente a la fecha de contratación, un monito, base para la computadora, un par de headsets, teclado y mouse. Para puestos donde no es necesario tanto rendimiento en las computadoras de entrega una MacBook Air, y para los internos se tienen computadoras Dell con Ubuntu como sistema operativo. Cada planta está equipada con una impresora láser conectada a la red y pueden ser usadas por cualquiera empleado. Existen 4 redes inalámbricas, 2 dedicadas solamente al trabajo, una tercera para dispositivos personales, como smartphones, tablets, etc. y una última para invitados. Cada lugar de la oficina tiene la opción de conectarse a la red directamente a través de un cable UTP.

En cuestión de equipamiento intelectual, Nearsoft cuenta con una amplia colección de libros alojados en los diferentes libreros ubicados en los diferentes pisos de la oficina, también se cuenta con una amplia lista de suscripciones a cuentas que nos dan el acceso a libros, cursos y talleres en línea a cualquier miembro de la empresa.

2.1 CLIENTES DE LA EMPRESA

Nearsoft no se enfoca en conseguir clientes para un solo proyecto y luego dejar la relación ahí, el enfoque de la empresa es formar parte de su equipo de trabajo, donde se busca apoyar el crecimiento del cliente con la ayuda de los recursos que Nearsoft brinda.

Entre la gran cantidad de clientes que la empresa contiene, los principales se muestran en la figura 2.4.



Figura 2.4 Principales clientes de Nearsoft

Switchfly permite a las agencias de viaje, programas de fidelidad e instituciones de servicios financieros para ofrecer una excelente experiencia a sus clientes. Dentro de sus clientes, Switchfly tiene a American Airlines, United Airlines, Starwood Hotels, Intercontinental Hotels, American Express, PayPal y esto solo por mencionar solo algunos.

OpenTable es un sistema en línea para hacer reservaciones a restaurantes, con una cartera de clientes de 32,000 restaurantes por todo el mundo. Cada mes se encuentran ahí más de 16 millones con más de 30 mil millones de dólares en ventas.

YottaMark ofrece soluciones de rastreo y autenticación de alimentos, que promueven la confianza y crean valor para las empresas y los consumidores.

3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS

A continuación se listan las herramientas principales utilizadas durante este proyecto, sean con las tareas programadas y también en proyectos de los cuales forme parte, como el Nearsoft Training.

- **Integrated Drive Electronics (IDE).** Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word. [1]
- **IntelliJ IDEA.** IntelliJ IDEA es un IDE para Java creado por Jet Brains del que existen dos distribuciones: Community Edition y Ultimate. [2]
- **Editor de Texto.** Un editor de texto es un programa informático que permite crear y modificar archivos digitales compuestos únicamente por textos sin formato, conocidos comúnmente como archivos de texto o “texto plano”. [3]
- **Atom.** Es un editor de texto moderno y de código abierto, orientado al desarrollo de aplicaciones; una herramienta desarrollada por Github. Este puede ser personalizado a las necesidades de cada quien. [4]
- **Sistemas de control de Versiones.** El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Cualquier tipo de archivo que encuentres en un ordenador puede ponerse bajo control de versiones. [5]
- **GIT.** Es un software de control de versiones diseñado por Linus Torvalds (creador de Linux). Fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. [6]

Algunas de las características más importantes de Git son:

- ❖ Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
 - ❖ Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
 - ❖ Gestión eficiente de proyectos grandes.
 - ❖ Realmacenamiento periódico en paquetes.
- **Java.** Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes. [7]
 - **Spring.** Es una plataforma que nos proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java. Spring maneja toda la infraestructura y así te puedes centrar en tu aplicación. Spring se encarga de unir todos los componentes de la aplicación, maneja su ciclo de vida y la interacción entre ellos. [8]
 - **Inversión de control.** Inversión de control es un patrón que permite revertir el flujo de control de un programa. Esto va desde el momento en que construimos una aplicación gráfica en la cual asociamos algunas de nuestras funciones con los eventos ocurridos desde la interfaz, es decir estamos cediendo el control de la ejecución de nuestro código a un tercero hasta el momento en que cedemos a un tercero el armado y conexión de nuestros componentes (instancias o módulos). [9]
 - **Inyección de dependencias.** La inyección de dependencias consiste en desacoplar nuestro código por medio de delegar a terceros la responsabilidad

de proveernos de las clases que dependemos para que el código que escribimos funcione. [10]

- **Angular JS.** Es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje Javascript con programación del lado del cliente. [11]
- **Yeoman.** Yeoman es un conjunto de herramientas construidas sobre Node.js y que lleva a cabo una serie de tareas para facilitar las tareas a la hora de desarrollar nuestras aplicaciones. Con unos pocos comandos, Yeoman se encargará de crear esqueletos para aplicaciones, comprimir imágenes o concatenar unos pocos CSS por poner un ejemplo. Funciona para cualquier lenguaje de programación. [12]
- **JHipster.** JHipster es una herramienta de generación basada en Yeoman generator para generar aplicaciones Web Spring Boot + AngularJS. [13]
- **Base de datos.** Es un sistema que almacena datos que están relacionados. Componentes de una Base de Datos:
 - Hardware: constituido por dispositivo de almacenamiento como discos, etc.
 - Software: que es el DBMS o Sistema Administrador de Base de Datos.
 - Datos: los cuales están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información. [14]

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se este manejando, o la utilidad de la misma. [15]

- **SQL (Structured Query Language).** Es un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla. Las consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y más. [16]
- **PostgreSQL.** Es un Sistema de gestión de bases de datos relacional orientado a objetos y libre. Como muchos otros proyectos de código abierto,

el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales.

- **Liquibase.** Liquibase es una librería open-source basada en Java que se encarga de la gestión y aplicación de cambios que se producen en nuestra base de datos independientemente de cual sea la que se use (base de datos relacionales). La idea central de esta librería es que cada uno de los miembros del equipo puedan contar con la información relacionada con los distintos cambios que sufrió la base de datos, quien fue la persona que los realizó y en que fecha o versión fueron hechos. [18]
- **Bootstrap.** Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. [19]
- **Mahout.** Es una librería de Machine Learning escrita en Java y optimizada para funcionar sobre Hadoop y tiene un gran número de algoritmos implementados para trabajar sobre las tres Cs del Machine Learning: filtración Colaborativa, Clusterización y Clasificación. [20]
- **Maven.** Maven es una herramienta open-source, que se creó con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente). [21]
- **Selenium.** Selenium es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas. [22]
- **WebRTC (Web Real-Time Communication).** Es una API que está siendo elaborada por la World Wide Web Consortium (W3C) para permitir a las aplicaciones del navegador realizar llamadas de voz, chat de vídeo y uso compartido de archivos P2P sin plugins. [23]

- **JUnit.** JUnit es un framework java que permite la realización de la ejecución de clases de manera controlada, para poder comprobar que los métodos realizan su cometido de forma correcta. [24]
- **Scrum.** 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. [25]
- **Continuous Integration (CI).** Es un modelo informático que consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes. [26]

4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS

En este apartado se presenta más a detalle las actividades que se estuvieron realizando durante las diferentes fases del Nearsoft Academy durante la prolongación de los 3 meses. Debido a una oportunidad que se presentó en el transcurso del primer mes, se tuvo que hacer un pequeño cambio en el orden de las fases originales del programa. Las actividades planeadas para el segundo y tercer mes cambiaron de orden, dejando así las contribuciones al código abierto al segundo mes y el proyecto desde cero se realizó en el tercer mes.

4.1 FASE 1. RESET

Durante el primer mes de mis prácticas profesionales realizadas dentro de las oficinas de Nearsoft, dentro del programa Nearsoft Academy se llevaron a cabo varias actividades para eliminar malos hábitos que teníamos y resolver la mayoría de las dudas con las que entramos a este programa. Además se realizan actividades para mejorar el trabajo en equipo y la comunicación.

Al ingresar como practicantes, a cada uno se nos asignó un mentor, el cual nos ayudó con las actividades que estábamos realizando, resolver dudas y recibir retroalimentación de nuestro desempeño dentro de la empresa. Durante este periodo, cada semana se nos asignaban leer lecturas con temas variados, como innovación, un libro sobre como hablar en público y transmitir tus ideas, metodologías de testing como test driven development, conceptos diversos como “mocks vs stubs”, manejadores de dependencias, como maven, npm y gulp, entre otros diversos tópicos. Una vez realizadas las lecturas cada semana se entregaba una tarea, en forma de ensayo a nuestro mentor del cual se nos daba retroalimentación de nuestro progreso.

Personalmente disfrute mucho del libro “The Pragmatic Programmer” de Andy Hunt y Dave Thomas, un libro que aunque casi tiene 20 años en el mercado (figura 4.1),

es una excelente base para cualquiera que quiera ser desarrollador. Es cierto que viene con muchos fragmentos de código obsoleto y platica de tecnologías inservibles a la fecha, pero no todo en la industria es el tener conocimiento técnicos, mucho tiene que ver lo que se toca en el libro, como el título de cada capítulo del libro, por ejemplo el ser cuidadoso con lo que haces y el valor que eso tiene, no dejar ventanas rotas, una analogía que nos hace ver como dejar huecos donde no den ir, puede hacer que todo se venga abajo.

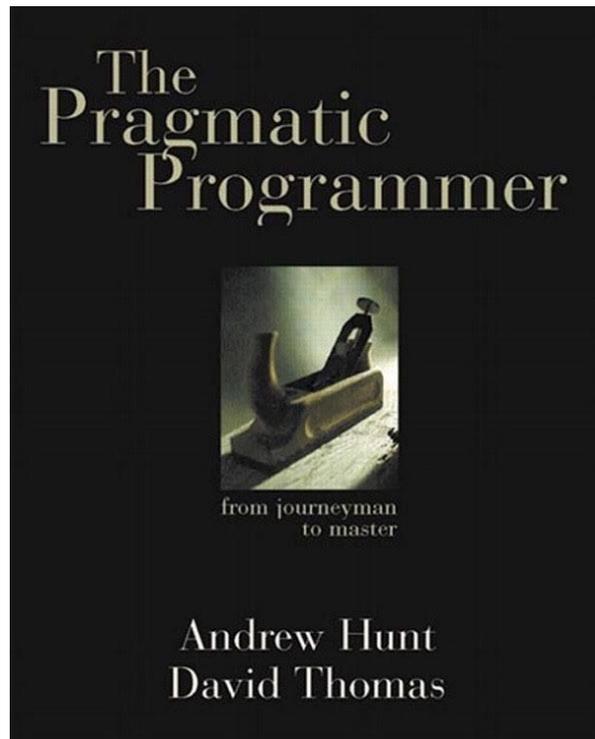


Figura 4.1 Portada de *The pragmatic programmer*

Además de las lecturas y videos, habían tareas extras, la segunda semana tuvimos que realizar 3 ejercicios en 3 lenguajes de programación diferentes (elixir, F# & ECMAScript 6). Cada uno de ellos implicó retos diferentes, como el uso de expresiones regulares, recorrer colecciones y realizar acciones sobre sus datos, etc. La 3er semana realizamos un ejercicio sobre Big Data, en el cual debíamos implementar un sistema recomendador de películas usando la librería Mahout, de apache. Este fue alimentando con un archivo de más de 8 millones de registros y al final arrojaba resultados en base a las coincidencias que se encontraba en el archivo.

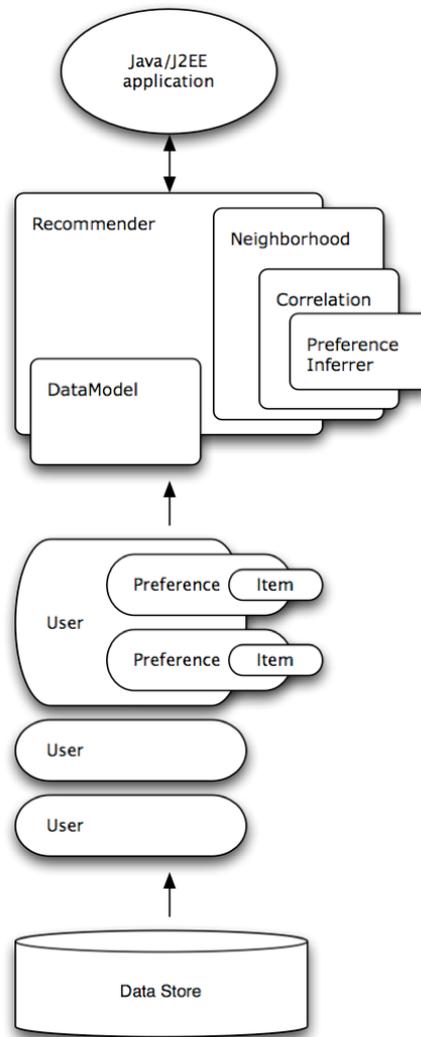


Figura 4.2 Arquitectura del sistema recomendador con el uso de Mahout

La 4ta y ultima semana debíamos generar ideas y crear un “pretotipo” de ella. Un pretotipo consiste en implementar algo sencillo para ver si la idea que tenemos en mente va a tener éxito, ver cómo se comporta la gente y ver como es la reacción que se tiene ante ella, puede ser aceptada y seguir buscar hacer la implementación real o puede ser un fracaso y lo ideal sería no continuar con ella. En mi caso, tuvimos la idea de hacer un sistema que por medio de una petición por tu teléfono este avisara si habían o no habían lugares en el estacionamiento de la empresa. El pretotipo que implementamos fue hacer un grupo dentro de la plataforma de Whatsapp y agregamos a alrededor de 30 personas ahí, cuando ellos se acercaban nos hacían una petición y nosotros dábamos un vistazo al estacionamiento y avisábamos si habían o no lugares disponibles, como podemos ver en la figura 4.3,

para que no perdieran tiempo buscando y fueran a un lugar directo donde sí podían encontrar.



Figura 4.3 Pretotipo de identificación de espacios en estacionamiento

Finalmente me agregaron al equipo de contribuidores de un sistema interno de Nearsoft, conocido como "Training" que tiene como función llevar un registro de los talleres que se realizan ahí dentro, donde la gente se puede inscribir, pedir talleres y presupuesto para algún entrenamiento, libro o curso, todo con el fin de mejorar las habilidades técnicas de los empleados de Nearsoft. Para comenzar escribí algunas pruebas faltantes y refactorización, esto con la finalidad de introducirme a la plataforma de jHipster, que es la utilizada en este sistema. En la figura 4.4 podemos apreciar como luce la página principal del sistema desde el punto de vista de un usuario.

training.nearsoft.com/#/

nearsoft training v1.0.0-RC1

Home View Cristian

Home

Welcome, Nearsoft Training!

Training requests

2 Active requests need your attention

Name	Requested Budget In USD	Start Date	Status	Created Date	Assigned To	Requested By
Certificado Profesional en Consultoría en Design Thinking	\$1,000.00	10/1/16 12:00 PM	Needs feed from Reviewers	2016-09-28T00:32:19.194Z		lortiz@nearsoft.com
Intelligent Teams & Networks	\$750.00	11/21/16 12:00 AM	New	2016-10-10T19:07:31.066Z		jrodriguez@nearsoft.com

+ Create a Training Request

Upcoming Sessions Next 10 days

Not enrolled yet

Session Type	Title	Starts at (tz America/Hermosillo)	Location
Workshop	Usability Testing Workshop Sold out	Oct 11, 2016 1:00:00 PM	Don Mina (HMO)
Workshop	CUU-SASS	Oct 12, 2016 9:00:00 AM	Chihuahua (CUU) enroll

Figura 4.4 Página inicial del sistema training

4.2 FASE 2. CONTRIBUIR AL CÓDIGO ABIERTO

Durante el segundo mes, las actividades se enfocaron en la contribución al código abierto, debido, esto con la finalidad de adaptarnos a leer el código de otras personas, mejorar nuestras habilidades técnicas y finalmente formar parte de esta gran comunidad que nos brinda un gran número de herramientas y de forma gratuita para la industria del desarrollo de software.

Para contribuir a los proyectos de código abierto es necesario hacer un “fork” del repositorio original, que es equivalente a sacar una copia hacia mi cuenta de github, después se clona el repositorio a forma local para poder hacer los cambios necesarios dentro del código y una vez que el problema fue arreglado o la nueva funcionalidad está lista y probada, el código vuelve a ser subido al fork creado y proseguimos haciendo un “Pull Request”, donde se hace la solicitud de que los colaboradores y miembros del proyecto revisen tus cambios y una vez que den el visto bueno, serán fusionados los cambios con el código original.

A lo largo de este periodo intente trabajar con algunos proyectos, enfocados en los lenguajes java y javascript, ya que son los dos lenguajes que mejor domino en la actualidad. Los proyectos que puse a correr e intenté resolver “issues” fueron los siguientes:

- Spring-framework (Java)
- Selenium (Javascript & Java)
- Angular-material (Javascript)
- Angular.js (Javascript)
- Bootstrap (CSS & Javascript)

Además de trabajar con estas contribuciones aprendimos y mejoramos nuestras habilidades de Git, que es el manejador de versiones más utilizado actualmente por la comunidad de desarrolladores y con el que se manejan un gran número de repositorios. En nuestro caso buscamos proyectos alojados en Github, que es el cliente remoto para git, ya que ahí se alojan bastantes proyectos de código abierto, los cuales nos servirán para poder entender cómo se manejan los proyectos con un cliente real.

Personalmente no tuve ningún Pull Request aceptado, debido a que la issues que seleccione no eran para mi nivel. Lo más cercano que estuve fue con *Bootstrap*, el issue consiste en que un modal seguía siendo visible aun cuando se desapareciera el elemento al cual era añadido se desapareciera. Fui capaz de arreglar este issue poniendo un listener que escuchara siempre al DOM y cuando ocurriera un cambio (como remover un elemento) revisara si no tenia algun modal agregado para que lo quitara junto con el. El detalle aquí fue que no cumplí con los estándares del código que se estaban llevando.

En este tiempo colaboré con la migración de algunos datos faltantes, que se encontraban en varios documentos dispersos, agregándolos con la ayuda de liquibase, una librería que se encarga de revisar, manejar y aplicar cambios a la base de datos

4.3 FASE 3. CREAR UN PROYECTO DESDE CERO

Durante este, el tercer el objetivo principal fue el de construir un proyecto desde cero, en nuestro caso el proyecto intentaría conectar la oficina de Chihuahua con la de Hermosillo por medio de streaming continuo. La idea original consistía en abrir una ventana entre las oficinas mencionadas, como podemos observar en la figura 4.5, pero finalmente se optó por crear un “portal” entre ellas.

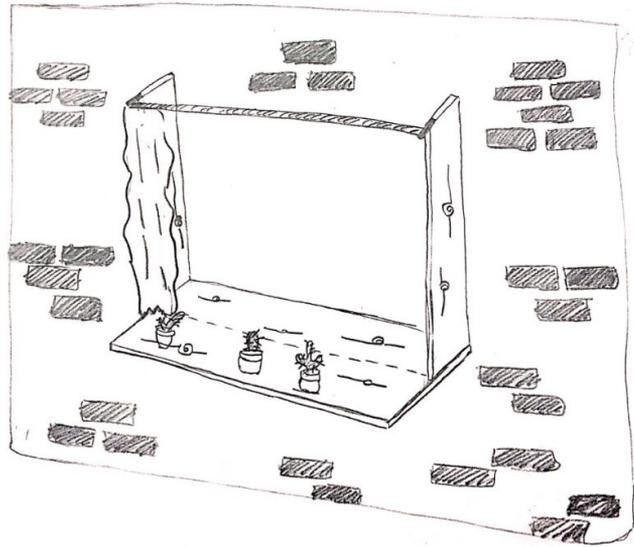


Figura 4.5 Diseño original de la ventana

Las actividades eran variadas en este proyecto, ya que el proyecto involucró hardware y software. Todo el equipo colaboró en el diseño del portal y cómo sería la interacción con el. Se nos otorgó un presupuesto con el cual debíamos cubrir todas las necesidades que este portal requería. Las tecnologías principal que se utilizó para el desarrollo del software fue WebRTC que nos brinda comunicación en tiempo real de forma relativamente sencilla, siendo aquí el área donde colaboré principalmente. En la figura 4.6 se puede ver una de las primeras pruebas que realizaron los compañeros que se enfocaron al hardware, donde realizaban pruebas con una cámara GoPro conectados a una Raspberry.



Figura 4.6 Pruebas con Raspberry Pi y Gopro

A lo largo de este periodo aprendimos a colaborar como equipo y cómo superar los problemas que se presentaban y como salir adelante de las limitantes que se nos presentaron, como falta de tiempo y presupuesto. El resultado final de cada oficina puede ser visto en la figura 4.7, donde vemos de lado izquierdo el portal Hermosillo que fue ubicado en la cocina y del lado derecho vemos el portal de Chihuahua, alojado en el cuarto juegos. En la figura 4.8 podemos ver el portal en función desde un punto de vista de Hermosillo.



Figura 4.7 Portal finalizado

(izquierda Hermosillo, derecha Chihuahua)

En lo que se daba el desarrollo del portal entre las oficinas, mis actividades como colaborador en el sistema de Training fueron creando un servicio que se encargaba de enviar correos con interés a diferentes usuarios con roles específicos. Por ejemplo el enviar un correo de confirmación a un usuario que se registraba a un taller o notificar a un usuario que había solicitado un presupuesto, si este era aprobado o rechazado.

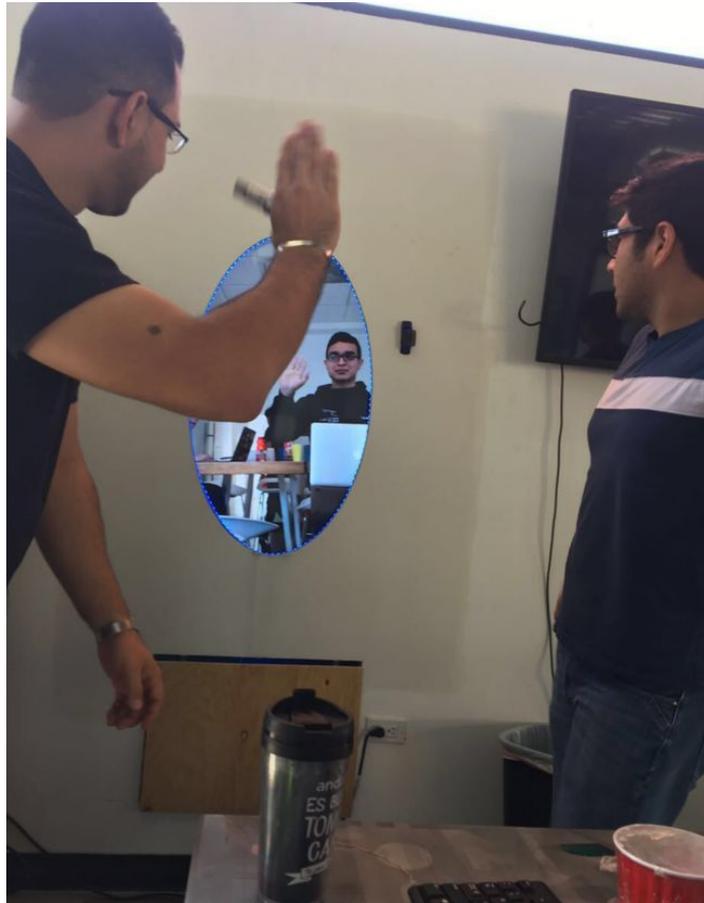


Figura 4.8 Portal funcionando con perspectiva de la oficina de Hermosillo

5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA

A continuación se describe cual ha fue la valoración personal sobre las etapas del Nearsoft Academy una vez que han sido concluidas, acerca del proyecto, como ha sido la experiencia de formar parte de la comunidad de Nearsoft, las actividades realizadas durante el transcurso.

5.1 ANÁLISIS GENERAL DEL PROYECTO

El proyecto del Nearsoft Academy en lo general ha sido una de las experiencias más gratificantes que he vivido durante mi vida profesional, ya que a lo largo de todas sus etapas el conocimiento de áreas muy diversas nunca fue interrumpido. El proyecto abarca todos los ámbitos que un desarrollador de software lleva a cabo en su día normal de trabajo, como el manejo de herramientas de trabajo indispensables para la colaboración con otros, como lo es git, también se te enseña a comunicar con los demás tus problemas, inquietudes o desacuerdos de manera adecuada sin interferir con el flujo de trabajo.

A medida en que se desarrolló el proyecto cada vez captaba mejor la importancia del porqué la existencia de algunas etapas o tareas, que personalmente en un principio no tenían sentido, y después ver cómo todo se unifica para lograr los objetivos que se señalaban en un principio.

5.2 ANÁLISIS DE LOS OBJETIVOS DE LAS PRÁCTICAS

Como ya mencioné anteriormente los objetivos son muy completos, ya que con una serie de objetivos bien definidos, se logra un objetivo final y el deseado para todos, ellos encuentran empleados los cuales son moldeados desde el comienzo de su vida profesional y el interno logra cumplir con su objetivo de conseguir un trabajo. Los objetivos intermedios, como la creación de un proyecto guarda objetivos los

cuales tienen que ser descubiertos, ya que no son explícitamente mencionados, en este caso en particular no se interesa terminar el proyecto y tener un producto finalizado, sino el objetivo aquí es darnos cuenta de las debilidades que tenemos personalmente y como enfrentarnos nosotros mismos para superar esos bloqueos que muchas veces ni siquiera nos percatamos de su existencia.

Otro objetivo que me pareció muy interesante que sea tan tomado en cuenta es la comunicación y sobre cómo mejorarla. A lo largo de los 3 meses y en específico el primero se realizan diversas actividades con el fin de interactuar con otros miembros de la empresa con el fin de llegar a un resultado, pero el objetivo se va cumpliendo de forma indirecta.

5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS

Las actividades realizadas son muy completas, hablando técnicamente siempre son muy desafiantes y no dejan que uno baje el nivel, las actividades que se realizan provocan que el interno siempre esté con ganas de aprender más y más, con el fin de terminar el trabajo, haciéndolo cada vez mejor, con la herramientas adecuadas y con una mejor calidad.

El aumento gradual de dificultad de las actividades del Nearsoft academy es notoria y lo interesante aquí es ver cómo en cada etapa, personalmente uno puede observar claramente que el avance que tiene desde en el ambiente desde el momento que comenzó el programa hasta el día de hoy.

El hecho de no tener como objetivo principal el terminar siendo un sabio de la programación me parece muy inteligente, ya que al no ser la principal prioridad, estas habilidades se desarrollan siempre y cuando se estén cumpliendo con las actividades que se piden realizar.

6. CONCLUSIONES Y RECOMENDACIONES

Al concluir con mis prácticas profesionales pude notar que sin lugar a duda siempre hay que estar dispuestos a aprender, personalmente ya había contado con trabajos previos a mis prácticas profesionales con relación al desarrollo de software, pero donde trabajé previamente no se contaba con una actualización de las tecnologías que se utilizan ni se escogen las adecuadas para los requerimientos de sus clientes. Es bien sabido que el área de tecnologías de información está actualizándose día con día y no hay nada que lo detenga, así es que lo mejor es no quedarse varado en lo que conocemos hoy en día y abrir nuestros horizontes a nuevas tecnologías y áreas que abren muchas puertas.

A manera de conclusión me gustaría hacer notar que el programa del Nearsoft Academy es una gran oportunidad para realizar las prácticas profesionales si el interés es ser un desarrollador y no se cuenta con las suficientes habilidades técnicas como para conseguir cubrir una vacante así como así al ser recién egresado o estar por egresar, como lo fue en mi caso.

Como recomendación a los próximos practicantes me gustaría hacerles saber que no se queden con lo que la escuela les ofrece, busquen oportunidades afuera e intenten definir desde temprana edad estudiantil a lo que se quieren dedicar una vez concluida su carrera, no caigan en una zona de confort y esperen ver donde se pueden “acomodar” allá fuera, construyan su camino partiendo de los que les interesa y les gusta hacer.

6.1 RECOMENDACIONES AL PROGRAMA DE ISI

Las recomendaciones que tengo para el programa de nuestra carrera sería solamente que se mantuviera más actualizada, hay tecnologías que se toman en ciertas clases que ya ni siquiera están en funcionamiento, y la mayoría de ellas ya no son utilizadas por la industria. Es muy frustrante saber que lo que aprendiste en la escuela no sirve de nada allá afuera, pero gracias a la capacidad de algunos maestros, que ven más allá del solo enseñar y optan buscar el autodidactismo del alumno, algunos logran sobresalir dentro y fuera de la escuela.

Sinceramente me gustaría que el programa fuera más práctico, en mi caso que me incliné por el desarrollo de software, podría decir que se podría ver más sobre desarrollo web, no solo de sitios estáticos sino de sistemas completos que son administrados en la nube. Hace ya algunos años que es el mejor enfoque para trabajar, es cierto que no es un total absoluto, pero es para donde va el futuro.

Propondría enseñar mejor las bases para cualquier área, como algoritmos, estructuras de datos, programación orientada a objetos, web, protocolos, bases de datos desde álgebra relacional, así como en el área de redes. Una vez que se tienen las bases sólidas lo que viene es complementario. También propondría hacer uso de sistemas de control de versiones como principal herramienta en todas las materias, ya que estos son indispensables para trabajar hoy en día y ese tema no se tocó a lo largo de toda la carrera.

Deberían enseñarse metodologías ágiles en las materias, ya que así es como se trabaja fuera, además de que enseña mucho sobre cómo trabajar en equipo y hacer que todos se interesen en colaborar.

7. REFERENCIAS

<http://nearsoft.com/faq/>

1 https://www.ecured.cu/IDE_de_Programaci%C3%B3n

2 <https://www.jetbrains.com/idea/>

3 https://es.wikipedia.org/wiki/Editor_de_texto

4 <https://atom.io/>

5

<https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones#Sistemas-de-control-de-versiones-locales>

6 <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>

7 https://www.java.com/es/download/faq/whatis_java.xml

8 <http://docs.spring.io/spring/docs/3.1.0.M2/spring-framework-reference/html/>

9

<http://stackoverflow.com/questions/9403155/what-is-dependency-injection-and-inversion-of-control-in-spring-framework>

10 http://www.tutorialspoint.com/spring/spring_dependency_injection.htm

11 <https://angularjs.org/>

12 <http://yeoman.io/learning/index.html>

13 <https://jhipster.github.io/presentation/#/>

14 <http://www.monografias.com/trabajos34/base-de-datos/base-de-datos.shtml>

15 <http://basededatos.over-blog.net/article-tipos-de-bases-de-datos-68319538.html>

16

<http://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas>

17 <https://www.postgresql.org/>

- 18 <http://www.liquibase.org/>
- 19 <http://getbootstrap.com/>
- 20 <https://mahout.apache.org/>
- 21 <https://maven.apache.org/>
- 22 <http://www.seleniumhq.org/>
- 23 <https://webrtc.org/>
- 24 <http://junit.org/junit4/>
- 25 <https://proyectosagiles.org/que-es-scrum/>
- 26 https://en.wikipedia.org/wiki/Continuous_integration