

UNIVERSIDAD DE SONORA



Reporte de prácticas profesionales -
Nearsoft Academy



Tomás Alfonso Ibarra Ballesteros

No. Exp. 211220504

CONTENIDO

INTRODUCCIÓN.....	2
DESCRIPCIÓN DEL ÁREA DE TRABAJO.....	3
DESCRIPCIÓN DEL PROYECTO.....	4
JUSTIFICACIÓN DEL PROYECTO.....	5
OBJETIVOS DEL PROYECTO.....	6
PROBLEMAS PLANTEADOS A RESOLVER.....	7
ALCANCES Y LIMITACIONES.....	8
Alcances.....	8
Limitaciones.....	8
FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS.....	10
PROCEDIMIENTOS EMPLEADOS Y ACTIVIDADES DESARROLLADAS.....	14
Fase de reset.....	14
Construcción de un proyecto desde cero.....	17
Contribución al código abierto.....	20
RESULTADOS OBTENIDOS.....	26
CONCLUSIONES Y RECOMENDACIONES.....	27
RETROALIMENTACIÓN.....	28
Fortalezas.....	28
Debilidades.....	28
Oportunidades.....	28
Recomendaciones al programa de ISI.....	29
REFERENCIAS.....	30

INTRODUCCIÓN

Hoy en día las empresas optan por tomar nuevas iniciativas sobre cómo poder dar un plus a sus servicios o a sus productos, es por eso que optan por generar nuevas opciones para brindar a los empleados una mejor capacitación para poder obtener mejores resultados a largo plazo, es por eso que empresas como Nearsoft deciden tener dentro de su visión tener proyectos tales como Nearsoft Academy.

Nearsoft academy es un plan donde los internos que quieren formar parte de esta empresa pasan alrededor de seis meses siguiendo un plan estructurado, en el cual trabajan en algunas áreas como: código libre, relación con clientes, agile, scrum, por mencionar algunas.

Todo esto es para preparar al interno para no toparse con una gran barrera al no saber como es el mundo laboral, por eso el plan actúa como un mediador para preparar al próximo *Nearsoftian*, donde aprenderá metodologías ágiles, practicará en proyectos de código de gran escala y aprenderá a utilizar software de revisión de versiones para poder colaborar y tener referencias para futuras entrevistas.

DESCRIPCIÓN DEL ÁREA DE TRABAJO

Nearsoft es una empresa dedicada al desarrollo de software, cuenta con clientes importantes dentro del comercio mundial tales como Switchfly o Points, cuenta con tres oficinas en la república mexicana, la principal está ubicada en Hermosillo, Sonora en el bulevar Dr. Antonio Quiroga, Quinta Emilia, C.P. 83214, las otras dos se encuentran en Chihuahua y en México D.F.

El CEO es Roberto Martínez, Nearsoft nació con la idea de superación y creación de una empresa amigable, competente y estable, teniendo clientes en Estados Unidos que tenga un ambiente donde se pueda trabajar sin preocupaciones, relajarse y disfrutar de lo que se hace, esta fué la idea principal de Roberto para crear su propia empresa, donde después llegaría Matt Pérez, actual COO de Nearsoft, quien apoyó los sueños y expectativas de Roberto.

Nearsoft es una empresa local reconocida por tener una gran cultura, gran rentabilidad, reconocida también por su calidad de trabajo y gran capacidad de crecimiento, se encuentra actualmente con el título del “Mejor lugar para trabajar” dentro del estado de Sonora.

DESCRIPCIÓN DEL PROYECTO

El proyecto, llamado Nearsoft Academy trata de preparar a los próximos *Nearsoftians* para encontrar un lugar dentro de la empresa y poder trabajar con uno de sus clientes, el proceso consta de una duración máxima de seis meses donde el interno aprenderá técnicas para trabajar en equipo, metodologías de desarrollo, no aprender a trabajar con un framework de un stack en específico si no aprender a adaptarse a cambios.

El primer mes consta solo de conocimiento en general de la empresa, su misión y su visión, como comunicarse dentro de ella para obtener los resultados y mostrar los resultados propios, dar a conocer también los proyectos internos que se desarrollan aquí, también se tocan temas de Big Data y estructura de datos, algoritmos y programación orientada a objetos.

El segundo mes consta de construir algo desde cero, donde se implementan metodologías ágiles de desarrollo para desarrollar un proyecto que no solo contenga software si no también hardware para mejorar las habilidades y entender las funcionalidades de los dispositivos que utilizan software.

El tercer mes consta de contribuir al código abierto, a algún proyecto de gran magnitud para formar currículum y para practicar como en realidad grandes equipos de trabajo colaboran entre ellos para construir una aplicación, se resuelven problemas y bugs de aplicaciones.

Los siguientes meses tratan de entrevistas con clientes propios de Nearsoft para tratar de obtener un puesto dentro de las opciones disponibles, todo esto para formar lazos amistosos entre los clientes y los nuevos empleados, se agendan videollamadas, pair programmings y daily meetings y otras actividades.

JUSTIFICACIÓN DEL PROYECTO

Para entender un poco más sobre el proyecto es necesario entender el porqué es realizado, este proyecto nace por la necesidad de contar con herramientas necesarias para trabajar de manera ágil y eficiente dentro de un equipo de trabajo de desarrollo, donde dentro del equipo se cuentan con desarrolladores, aseguradores de calidad entre otros, y es muy difícil poder trabajar de esta forma sin tener bases previas y solamente entrar de lleno a un proyecto, es por eso que Nearsoft Academy conlleva una serie de prácticas que ayudan al interno a formarse profesionalmente para poder dar el mejor trabajo posible desde un principio.

Dentro del proyecto no solo se trabajará con una tecnología en específico o algún lenguaje, si no que se tratarán de abarcar diferentes áreas tales como front-end, back-end, hardware, por mencionar algunas, también se trabajará en proyectos internos de Nearsoft como colaborador solucionando bugs o peticiones a los repositorios alojados en la cuenta de github personal y privada de Nearsoft que cuenta con mas de diez proyectos internos utilizados dentro de la comunidad de Nearsoft.

Pero como se dijo anteriormente el objetivo primordial de este proyecto es al final terminar dentro de los equipos de desarrollo de alguno de los clientes de Nearsoft indiferentemente del stack en el que se trabaje el proyecto, para esto es imprescindible crear una lógica avanzada y una capacidad de aprendizaje rápido y ágil.

OBJETIVOS DEL PROYECTO

Los principales objetivos del proyecto son:

- Comprender como los verdaderos equipos de desarrolladores trabajan continuamente.
- Aprender técnicas para agilizar el proceso de aprendizaje.
- Utilizar git como software de revisión de versiones para trabajar en conjunto con otros desarrolladores.
- Conseguir contribuciones a proyectos de código abierto creando Pull Request esperando que sean aceptados y les den estatus de Merge.
- Aprender a trabajar con máquinas virtuales y herramientas de software para la creación de las mismas.
- Desarrollar un proyecto utilizando hardware y software para su funcionamiento.
- Apoyar a los proyectos internos de la empresa solucionando bugs dentro de ellos.
- Contar con herramientas para pasar entrevistas de ofertas de empleo.
- Conocer cómo funcionan ciertas estructuras de datos para agilización de procesos.
- Resolver problemas de programación en lenguajes nunca antes vistos por los integrantes del Academy.
- Utilizar la metodología SCRUM y Agile para aprender a trabajar en equipo.
- Crear relaciones profesionales con clientes de Nearsoft.
- Conseguir empleo dentro de los equipos de desarrollo de los clientes de Nearsoft

PROBLEMAS PLANTEADOS A RESOLVER

Nearsoft cuenta actualmente con una demanda grande de empleados, por lo que Nearsoft Academy considera a las personas que ya se encuentran listas para trabajar dentro de un equipo de desarrollo para un cliente real, donde tenga que respetar el horario de las daily meetings, resolver las cartas asignadas y trabajando de manera conjunta con los demás integrantes del equipo. Es por esto que el programa dura tanto tiempo, para poder capacitar de manera óptima al propenso nuevo empleado.

Otro problema es generar una base sólida de currículum para tener evidencias de los conocimientos de la persona, esto se crea al contribuir con los proyectos de código abierto de gran magnitud, ya que esto genera una confianza en que esta persona sabe cómo trabajar dentro de un equipo porque es de esta manera como se trabaja en Nearsoft, por equipos de desarrollo, también se encuentra con el problema de agilidad de aprendizaje es por esto que el primer mes del programa consta de varias lecturas enfocadas a la enseñanza de cómo aprender a aprender, metodologías ágiles, nuevas tendencias, big data, estructura de datos, por mencionar algunas, ya que independientemente del lenguaje esto no cambia, cambia la forma en la que es declarada más no cambia la interpretación ni la lógica.

El siguiente y último problema es formar a la persona con la cultura que se vive dentro de Nearsoft ya que es una parte importante dentro de la empresa, tanto que hasta está indicada dentro de la Visión de ella, siendo así una parte imprescindible para los empleados y para la convivencia dentro de los equipos, ayudando a la comunicación efectiva y rápida.

ALCANCES Y LIMITACIONES

Alcances

El programa consta como alcance final conseguir un contrato dentro de Nearsoft, para trabajar con alguno de sus clientes, pero también existen otros alcances menores que ayudan a alcanzar el final, los cuales son los siguientes:

- Aprobar el curso de *Learning how to learn*.
- Aprender a trabajar con Agile y SCRUM.
- Hacer mínimo una contribución a un repositorio de código abierto de gran magnitud.
- Crear un proyecto desde cero.
- Tener múltiples entrevistas con clientes potenciales.
- Adaptarse a la cultura de Nearsoft.
- Conocer a la comunidad de Nearsoft y generar amistades.

Limitaciones

Durante mi estancia en la empresa se presentaron diversas limitaciones que impactaron tanto en la calidad de mi trabajo como en el tiempo de entrega o terminación del proyecto, sin embargo, a pesar de todo se pudo lograr lo que la empresa solicitaba.

A continuación se listan algunas de las limitaciones que se presentaron en el desarrollo del proyecto:

- Falta de claridad en cuestión de magnitud de proyectos de código abierto.
- Falta de disposición del encargado del programa, ya que se encontraba consiguiendo ofertas de clientes para aceptar internos dentro de sus equipos.

- Contar con elementos del equipo que residen en Chihuahua, siendo la comunicación no tan constante como se desearía.
- Desconocimiento total de la cultura de Nearsoft.
- Tener poca experiencia con trabajo de equipo dentro de equipos con una cantidad de mas de 20 personas.
- Conocimiento nulo de los lenguajes de programación manejados dentro del programa tales como F# y Elixir.
- Competencia elevada para ganar un lugar dentro de los grupos de desarrollo.
- Nula experiencia en trabajos profesionales.
- Poca experiencia en el manejo de git para colaboración en equipos de mas de 100 mantenedores.
- Tiempo limitado de mi mentor asignado por el programa para ayudarme a sobrepasar los problemas o estancamiento al momento de trabajar sobre algún problema.

FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS

- **Atom:** Editor de texto de código libre para OS X, Linux y Windows con soporte para plugins escritos en Node.js y control git embebido, desarrollado por GitHub. Atom es una aplicación de escritorio construida con tecnologías web. Escrito en CoffeeScript y Less. [1].
- **intellij IDEA:** Entorno de Desarrollo Integrado (IDE) de Java para el desarrollo de sistemas computacionales. El IDE provee herramientas de build/packaging para la integración como grunt, bower, gradle y SBT. Soporta los sistemas de control de versiones como GIT, Mercurial, SVB, Bases de datos como SQL Server, ORACLE, PostgreSQL, y MySQL pueden ser accedados directamente desde el IDE. [2].
- **Java:** Es un lenguaje de programación y plataforma informática lanzada por Sun Microsystems en 1995. Su sintaxis fue mayormente basada en C y C++. Las aplicaciones Java son usualmente compiladas a bytecode que pueden correr en cualquier Máquina Virtual de Java (JVM). [3].
- **PHP:** Acrónimo recursivo de PHP: Hypertext Preprocessor, es un lenguaje de código abierto e independiente de plataforma, muy popular especialmente adecuado para el desarrollo Web que puede ser incrustado en HTML. El código es ejecutado en el servidor, generando HTML y enviándolo al cliente. [4].
- **Ruby On Rails (ROR):** Es un entorno de desarrollo web de código abierto para construir aplicaciones web que acceden a bases de datos. Framework de modelo vista controlador (MVC) con un conjunto de librerías,

automatismos y convenciones destinados a resolver los problemas más comunes a la hora de desarrollar una aplicación web. [5].

- **Javascript:** A veces abreviado como JS, es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase más conocido como el lenguaje de script para páginas Web, pero también usado en muchos entornos sin navegador, tales como node js o Apache CouchDB. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa [6].
- **F#:** Es un lenguaje de programación que proporciona compatibilidad con la programación funcional así como con la programación convencional orientada a objetos e imperativa (de procedimiento). [7].
- **Go:** Lenguaje de programación de código abierto creado por Google en 2007. Diseñado principalmente para la programación de sistemas, es un lenguaje compilado, de tipado estático como Algol y C, con recolector de basura, características de seguridad de memoria y reflexión en tiempo de ejecución. [8].
- **Elixir:** Lenguaje de programación funcional, concurrente con objetivos generales que corre en la máquina virtual Erlang. Elixir se construye en la parte superior de Erlang para proveer aplicaciones imparables, distribuidas, tolerante a fallas, suaves en tiempo real pero a la vez extiende su soporte a la meta programación con macros y polimorfismo vía protocolos. [9].
- **Spring:** El framework Spring es una plataforma de Java que provee soporte de infraestructura integral para el desarrollo de aplicaciones en Java. Spring maneja la infraestructura para que el desarrollador pueda enfocarse en su aplicación. [10].

- **Scrum:** Es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. [11].
- **Git:** Es un software de control de versiones distribuido, pensando en la eficiencia y confiabilidad del mantenimiento de versiones de aplicaciones que contienen un gran número de archivos de código fuente. Git almacena la información como instantáneas del proyecto a lo largo del tiempo. [12].
- **Maven:** Es una herramienta de gestión de proyectos que abarca un modelo de proyecto objeto, un conjunto de normas, un ciclo de vida del proyecto, un sistema de gestión de la dependencia y lógica para la ejecución de los objetivos de plugin en las fases definidas en un ciclo de vida. Y en términos más sencillos es una herramienta de *build* para artefactos desplegables de código fuente. [13].
- **Vagrant:** Es una herramienta para construir entornos de desarrollo completos. Con un flujo de trabajo fácil de usar enfocado en la automatización. Vagrant disminuye los tiempos de configuración del entorno de desarrollo e incrementa la paridad del desarrollo / producción. [14].
- **Docker:** Es una plataforma abierta para construir, enviar y correr aplicaciones distribuidas. Utilizado por las organizaciones para acelerar el proceso de desarrollo y despliegue de aplicaciones. [15].
- **VirtualBox:** Software de virtualización para arquitecturas x86 y AMD64/Intel64. Una de sus características es su alto rendimiento y que es la única solución profesional en su categoría disponible como software de código abierto. [16].
- **Discourse:** Proyecto de código abierto, desarrollado en Ruby. Es un nuevo tipo de foro, recién surgido, cuya meta es proveer una plataforma deseable para la siguiente década de foros. Discourse intenta enfocarse más que

nada en la conversación. Además es posible acceder a una conversación usando alguna red social como Google, Facebook, Twitter, etc. [17].

- **Helpy:** Es una moderna aplicación de ayuda desarrollada en Ruby. Algunas de sus características son base del conocimiento multilingüe, discusiones en comunidad y tickets privados integrados con correo electrónico. [18].
- **Travis CI:** Es un servicio de integración continua usado para hacer build y probar proyectos de software almacenados en GitHub. Provee despliegues personalizados de una versión propietaria en el propio hardware del cliente. Enfocado a la comunidad de código abierto. [19].

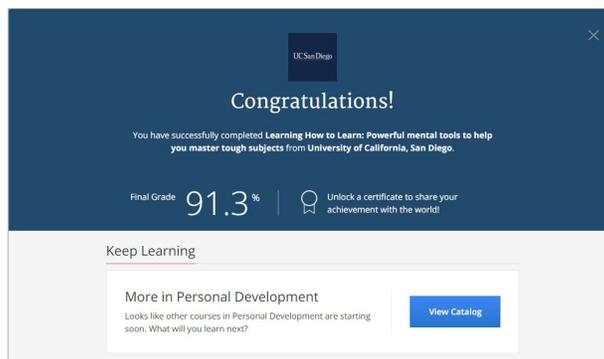
PROCEDIMIENTOS EMPLEADOS Y ACTIVIDADES DESARROLLADAS

1. Fase de reset

El primer mes constó de una fase de reset, donde los encargados del programa se dan la tarea de cambiar la mentalidad de los miembros del programa de pensar que en realidad tenemos las habilidades necesarias para trabajar dentro de un equipo, en esta fase de miran temas tales como estructuras de datos, SOLID, Big O notation, algoritmos comunes y programación orientada a objetos, se toman clases diarias de duración de una hora mirando dichos temas.

Dentro de esta etapa también se realizan lecturas que indican las tendencias en tecnología, cursos de aprendizaje tales como Learning how to learn impartido en Coursera, una plataforma dedicada a la toma de cursos, dentro de este curso se definen estrategias y métodos de aprendizajes que ayudan al entendimiento de cómo el cerebro trabaja al aprender algo nuevo, al final del curso dan una calificación.

Figura 1.1. Calificación del curso Learning how to learn de Coursera.



Dentro de este mes también se realizó la resolución de problemas diseñados en por Google code jam, una plataforma que cuenta con distintos problemas, dándote una entrada y tu tienes que generar tu salida utilizando la lógica que maneja el problema y la plataforma analiza tu código y tu salida y dependiendo de esto marca como resuelto o no tu problema, se realizó la resolución de 3 problemas diferentes en 3 lenguajes diferentes, teniendo así 9 soluciones en diferentes lenguajes: Elixir, F# y ECMAS6, a continuación muestro dos soluciones a un problema en dos lenguajes diferentes, Elixir y F#.

Figura 1.2 Solución al problema de Reverse Words en F#.

```
ReverseWords.fs Raw
1 open System
2 open System.IO
3
4 let read = File.ReadAllText("B-small-practice.in")
5 let lines = read.Split ['\n']
6
7 for i in lines do
8     let rev = Array.rev (i.Split [' '])
9     let words = String.concat " " rev
10    printf "%s" words
11    printf "%c" '\n'
```

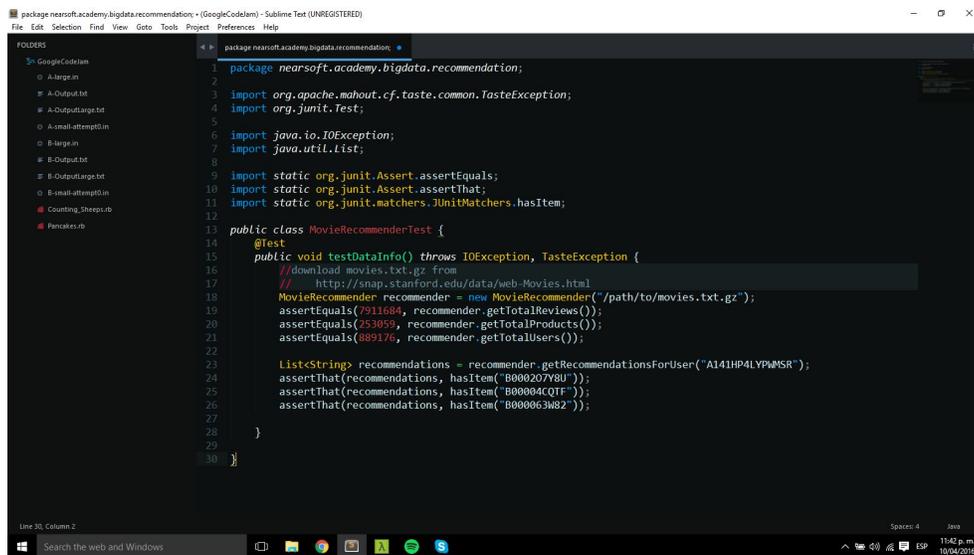
Figura 1.3 Solución al problema de Reverse Words en Elixir.

```
ReverseWords.exs Raw
1 raw = File.read! "B-small-practice.in"
2
3 list = String.split(raw, "\n")
4 list_wthoNo = List.delete_at(list, 0)
5
6 list_wthoNo |> Enum.with_index |> Enum.each( fn {n, i} ->
7     words = Enum.reverse(String.split(n, " "))
8     IO.puts "Case #<>Integer.to_string(i+1)<>": "<>Enum.join(words, " ")
9 end)
```

Estos ejercicios fueron puestos para agilizar la lógica dentro de los integrantes del equipo de Nearsoft Academy. Dentro de este mes también se observaron bastantes videos con temáticas de integración de equipos, conferencias de Google I/O, Machine learning y otros temas de interés.

Se creó también un recomendador de películas (fig 1.4) basado en un archivo de 8 millones de reviews, el propósito fue generar un listado de películas para tres usuarios dados, basándose en las películas anteriormente vistas, para esto se utilizó big data, data mining y otras técnicas.

Figura 1.4 Movie Recommender utilizando Mahout, Maven y Unit Testing.



```
package nearsoft.academy.bigdata.recommendation;

import org.apache.mahout.cf.taste.common.TasteException;
import org.junit.Test;
import java.io.IOException;
import java.util.List;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertThat;
import static org.junit.matchers.JUnitMatchers.hasItem;

public class MovieRecommenderTest {
    @Test
    public void testDataInfo() throws IOException, TasteException {
        //download movies.txt.gz from
        // http://snap.stanford.edu/data/web-Movies.html
        MovieRecommender recommender = new MovieRecommender("/path/to/movies.txt.gz");
        assertEquals(7911684, recommender.getTotalReviews());
        assertEquals(253059, recommender.getTotalProducts());
        assertEquals(889176, recommender.getTotalUsers());

        List<String> recommendations = recommender.getRecommendationsForUser("A141HP4LYPW5R");
        assertThat(recommendations, hasItem("B000207Y8U"));
        assertThat(recommendations, hasItem("B00004CQ1F"));
        assertThat(recommendations, hasItem("B000063W82"));
    }
}
```

Este mes se trató más de aprendizaje de tecnologías, estructuras de datos y metodologías de aprendizaje para después poder aplicar esto dentro de los siguientes dos meses más importantes que son la contribución al open source, fue un mes de mucho aprendizaje e integración entre el equipo de Nearsoft Academy.

2. Construcción de un proyecto desde cero

Dentro de esta fase se propuso un proyecto para que fuera realizado por los integrantes del Nearsoft Academy, la idea original fue la siguiente: Crear un portal entre dos oficinas de Nearsoft (CUU y HMO) que consta de un streaming de video “permanente” donde se transmita video y audio, pero no tiene que parecer una televisión o una pantalla haciendo streaming o una videollamada, si no que tiene que aparentar ser una ventana a otro lugar, en este caso Chihuahua.

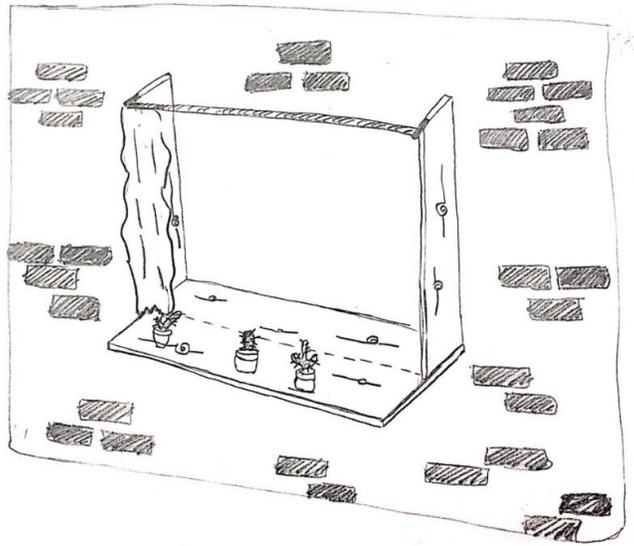
Al principio optamos por un diseño de una ventana, por lo que llamamos al proyecto Kitchen Window, este diseño constó de un marco de ladrillos, un marco más pequeño que cubriría la televisión, y tendría una repisa para colocar plantas o cualquier otro objeto decorativo para la simulación de la ventana, y el streaming sería reproducido por una Raspberry Pi y una cámara digital conectada a dicho dispositivo, teniendo corriendo el streaming en una red privada con un ancho de banda estático y dedicado entregado por los servidores de Nearsoft, en la siguiente figura podemos observar el prototipo del dispositivo (fig 2.1).

Figura 2.1 Raspberry Pi y Gopro.



De igual forma se hizo un bosquejo del diseño deseado para la realización de los marcos de madera y ladrillo, en caso de contratar a un carpintero para hacer el trabajo, esto se muestra en la siguiente figura.

Figura 2.2 Bosquejo de Kitchen Window.



Tras días de trabajo de estar testeando cámaras, sitios web dedicados al streaming directo de videos, utilizar cámaras IP, y otros dispositivos, una compañera del equipo se encontró con una idea que vendría a cambiar la perspectiva del proyecto, basándose en un juego de primera persona llamado Portal, encontramos un diseño futurista de lo que queríamos hacer (fig 2.3), utilizando facetime, una aplicación de la compañía Apple para sus sistemas operativos, utilizaron una pantalla para recrear los portales que se generan en el juego (uno azul para la entrada y uno amarillo para la salida), entonces optamos por tomar el camino de generar este proyecto de esa manera, se hizo la cotización de los equipos requeridos, pero de momento el proyecto está en estatus de espera

por motivos de finanzas, al ser dispositivos caros, no se puede hacer aún la compra de los dispositivos, teniendo señal de poder comprar dichos dispositivos se empezará con la realización y pruebas del proyecto.

Figura 2.3 Diseño del portal para el proyecto.



El diseño constará de una base móvil que se encontrará sosteniendo una pantalla dentro de ella, la base será una base rectangular, con una cámara incluida y una ranura en forma de óvalo con una tira de leds de color a su alrededor, simulando la entrada al portal, estando una estructura en cada oficina, de esta manera el portal podrá moverse y podremos interactuar con otra oficina de Nearsoft sin necesidad de agendar una videollamada, todo esto para aprender una cultura de la otra, ya que al tener dos oficinas de la misma empresa las culturas tienden a distanciarse, y para evitar esto es que se generó dicha idea. Fue necesario también dar una mini presentación a la empresa de cómo el proyecto funcionaría y vender la idea básicamente, para esto me basé en el libro de Resonate[20], que ayuda a la presentación de ideas y de cómo ganarte una audiencia a través de ciertos pasos.

3. Contribución al código abierto

Para el tercer mes de Nearsoft Academy entró en rigor la contribución de código abierto, para esta etapa la idea principal es contribuir a un proyecto que tenga gran magnitud, que sea reconocido al menos por la comunidad y que tenga vida, por tener vida me refiero a que los mantenedores y desarrolladores aún estén al tanto de la aplicación, no diariamente pero al menos semanalmente para así puedan revisar nuestras aportaciones y de ser posible merger nuestras soluciones con su código en producción.

Se nos dió una lista de proyectos ya trabajados por otras generaciones de Academy en caso de tener interés en una trabajar en ella, yo opté por buscar mi propia aplicación sin tomar en cuenta la lista, y me encontré primeramente con una aplicación llamada Helpy, dentro de ella contiene un archivo de configuración dónde tú defines tus parametros, servidor donde será alojado entre otras, este código contiene comentarios que ayudan a la configuración, dentro de ellos había demasiados errores de ortografía por lo que hice un Pull Request arreglando dichos problemas, el mantenedor miró mi solución (fig 3.1) y combinó mi solución con su código en producción, teniendo así mi primera pequeña contribución al código abierto.

Figura 3.1 Merge de mi solución de errores ortográficos en config.yml

Correct some typing mistakes. #84 Edit

Merged scott merged 3 commits into helpyio:master from tonisibarrab:master on 10 Mar

Conversation 3 Commits 3 Files changed 3

Showing changes from all commits - 1 changed file - +4 -4 Diff options ▾

```
8 config/settings.yml
@@ -10,7 +10,7 @@ site_url: http://support.yourdomain.com # This is the URL of your support site
10 site_name: Helpy Support # This is the name of your support site
11 site_tagline: Support and Helpdesk Software # This is the tagline of your support site
12 product_name: # This is the product name you are supporting
13 -support_phone: # This is support number to be displayed
14
15 # Analytics and Tracking
16 google_analytics_id: #Enter your analytics ID
@@ -21,17 +21,17 @@ send_usage_data: true
21
22 admin_email: inbound.support@yourdomain.com
23 from_email: inbound.support@yourdomain.com # This is the email that will be used for email replies (reply to)
24 -send_email: true #send email or not
25
26 mail_service: :mandrill # :sendgrid :cloudmailin, :postmark, :mandrill, :mailgun
27 smtp_mail_username: # Enter your SMTP email username
28 smtp_mail_password: # Enter your SMTP email password
29 mail_smtp: 'smtp.mandrillapp.com' #Enter the server address for your SMTP server
30 mail_port: 587 #Enter the port of your SMTP server
31 -mail_domain: #Enter your the heroku.com if you are on heroku, otherwise enter your domain
32
33 # OmniAuth Config
34 -# You must sign up with each oauth provider you wish to support and add the ID and secret
35 # for each below. See http://support.helpy.io/en/knowledgebase/11-Installing-Helpy/docs/19-Setting-Up-OAUTH-for-your-Helpy
36
```

Después de dos días de trabajo sobre esta aplicación decidí moverme a otra un poco más grande y viva, busqué una aplicación construida en Ruby on Rails ya que de momento me encuentro familiarizado con este framework y pensé que sería de ayuda conocer cómo funciona y tener un poco de dominio sobre el lenguaje y entonces encontré el proyecto en el que pasé este mes de contribución trabajando sobre este, dicho proyecto es Discourse, una aplicación creada por el co-fundador de Stack Overflow, una plataforma de ayuda a programadores reconocida mundialmente por sus soluciones y su comunidad.

Leí mucho código antes de meter mis manos en solucionar problemas de la plataforma, luego me di cuenta al cabo de 3 días aproximadamente, que es más fácil aprender a nadar aventándose al agua sin pensarlo, y me propuse a resolver un bug, el bug consistía en que dentro del foro de Discourse, no podías crear un hashtag con una longitud mayor de 50 caracteres después del “#”, dado que de

acuerdo a twitter, la longitud máxima de un hashtag es de 101 caracteres, así que solucioné ese bug, y al cabo de un día uno de los mantenedores contestó mi solución y dió *Merge* a esta, tal y como se muestra en la siguiente figura (fig 3.2).

Figura 3.2 Merge a solución dada para arreglar la longitud de Hashtag.

The screenshot shows a GitHub pull request interface. At the top, the title is "FIX: Length for hashtags, increased to 101 characters. #4081" with an "Edit" button. Below the title, it says "Merged" and "ZogStriP merged 1 commit into discourse:master from tomasibarrab:master 27 days ago". There are statistics for "Conversation 2", "Commits 1", and "Files changed 1". The main content area shows a comment from tomasibarrab: "I increased the char length for the hashtags to 101 characters with @tgxworld help, in order to solve this bug: [https://meta.discourse.org/t/hashtag-linking-to-categories-doesn-t-always-work/41003/7](\"https://meta.discourse.org/t/hashtag-linking-to-categories-doesn-t-always-work/41003/7\")". Below this is a commit message: "Fix length for hashtags, increased to 101 characters." with a green checkmark and "1542937". Then, a comment from discoursebot: "You've signed the CLA, tomasibarrab. Thank you! This pull request is ready for review." followed by a comment from tomasibarrab: "Ready to be merged @coding-horror @tgxworld @ZogStriP @eviltrout, can we review it?". At the bottom, it says "ZogStriP merged commit fb6bd10 into discourse:master 27 days ago" with "1 check passed" and buttons for "View details" and "Revert". On the right side, there are sections for "Labels" (None yet), "Milestone" (No milestone), "Assignee" (No one assigned), "Notifications" (Unsubscribe button), and "3 participants".

Después de esta solución empecé a estar revisando los bugs reportados por los usuarios de la aplicación durante todo el mes, las soluciones las realicé en Atom mientras que los Pull Request los hacía desde Git, dentro del mes abrí 9 Pull Request, de los cuales 5 fueron cambiados a estatus de *Merge*, los otros siguen en estatus de open y tal vez nunca los revisaron o no les dieron seguimiento, a continuación explicaré un poco sobre los otros 4 bugs que solucioné.

El siguiente constó en una solución de Front-End para la sección de emblemas, Discourse cuenta con una sección de emblemas para sus usuarios donde cada emblema es recibida después de cierta acción, los emblemas son diferenciadas en 3 tipos, bronce, plata y oro dependiendo de tu cantidad de acciones realizadas, el

problema era en que los colores no se estaban respetando para los iconos con forma de corazón, el CSS del color de “Me gusta” de un corazón, que es rosa, se estaba viendo reflejado en los emblemas de la forma de corazón, por lo que se veían corazones rosas en la sección de emblemas, lo cual era incorrecto, solucione dicho problema modificando el CSS de la sección de corazones (fig 3.3) y mi solución fue aceptada, después hice un cambio menor en la estructura de un archivo explicativo donde un link directo no estaba siendo visto como un link si no como una línea de texto (fig 3.4) por lo que arregle eso y fue aceptada mi solución.

Figura 3.3 Solución de corazones rosas en la sección de emblemas.

```
6 app/assets/stylesheets/common/base/user-badges.scss
@@ -20,15 ~20,15 @@
20 }
21
22 &.badge-type-gold .fa {
23 -   color: #ff6700;
24 }
25
26 &.badge-type-silver .fa {
27 -   color: #0c0c0c;
28 }
29
30 &.badge-type-bronze .fa {
31 -   color: #cd7f32;
32 }
33 }
34

20 }
21
22 &.badge-type-gold .fa {
23 +   color: #ff6700 !important;
24 }
25
26 &.badge-type-silver .fa {
27 +   color: #0c0c0c !important;
28 }
29
30 &.badge-type-bronze .fa {
31 +   color: #cd7f32 !important;
32 }
33 }
34
```

Figura 3.4 Cambio de formato de link en VAGRANT.md

UPDATE: VAGRANT.md, link to become eviltrout admin. #4106

Edit

Merged ZogStrip merged 1 commit into discourse:master from tomasibarrab:vagrant.md 20 days ago

Conversation 2 Commits 1 Files changed 1

Showing changes from all commits 1 changed file

+1 -1 Diff options

```
docs/VAGRANT.md
@@ -94,7 +94,7 @@ In a few seconds, rails will start serving pages. To access them, open a web bro
94
95 If you want to log in as a user, a shortcut you can use in development mode is to follow this link to
96 log in as 'eviltrout':
97 -http://localhost:4000/session/eviltrout/become
98
99 You can now edit files on your local file system, using your favorite text editor or IDE. When you
100 reload your web browser, it should have the latest changes.
```

Mi última contribución aceptada fue el cambio de una condicional que estaba mal, dentro de la parte de traducción de lenguaje inglés a rumano, el problema estaba en que en el lenguaje rumano no se estaba agregando el artículo “de” que se tiene que poner después de tener veinte unidades de algo, ejemplo: se tienen 19 me gusta en una publicación, la traducción a romanesc sería “19 aprecieri”, pero cambia cuando se tiene 20 unidades, en 20 me gusta la traducción literal sería “20 de aprecieri”, y esta regla no se estaba respetando, fue así como llegué a que la conclusión estaba en el controlador del lenguaje rumano, incluí unos módulos a las condiciones para que funcionara (fig 3.5) y la solución fue aceptada.

Figura 3.5 Solución a la traducción rumana en la pluralización.

FIX: Romanian pluralization rules using modulus #4128

Edit

Merged SamSaffron merged 1 commit into discourse:master from tomasibarrabroman_plural 11 days ago

Conversation 4 Commits 1 Files changed 1

Showing changes from all commits 1 changed file

Diff options

```
3 config/locales/plurals.rb
@@ -79,7 +79,7 @@
79 ips => { :i18n => { :plural => { :keys => [:one, :other], :rule => lambda { [n] n == 1 ? :one : :other
80 } } },
81 :pt => { :i18n => { :plural => { :keys => [:one, :other], :rule => lambda { [n] [0, 1].include?(n) ?
82 :one : :other } } },
83 :pt_BR => { :i18n => { :plural => { :keys => [:one, :other], :rule => lambda { [n] n == 1 ? :one :
84 :other } } },
85 :ro => { :i18n => { :plural => { :keys => [:one, :few, :other], :rule => lambda { [n] n == 1 ? :one :
86 n == 1 && n <= 19 ? :few : :other } } },
87 :ru => { :i18n => { :plural => { :keys => [:one, :few, :other], :rule => lambda { [n] n % 10 == 1 && n
88 % 100 != 11 ? :one : [2, 3, 4].include?(n % 10) && ![12, 13, 14].include?(n % 100) ? :few : :other } } },
89 :se => { :i18n => { :plural => { :keys => [:one, :two, :other], :rule => lambda { [n] n == 1 ? :one :
90 n == 2 ? :two : :other } } },
91 :sn => { :i18n => { :plural => { :keys => [:one, :few, :many, :other], :rule => lambda { [n] n % 10 ==
92 1 && n % 100 != 11 ? :one : [2, 3, 4].include?(n % 10) && ![12, 13, 14].include?(n % 100) ? :few : n % 10
93 == 0 || [5, 6, 7, 8, 9].include?(n % 10) || [11, 12, 13, 14].include?(n % 100) ? :many : :other } } },
94 :sv => { :i18n => { :plural => { :keys => [:one, :two, :other], :rule => lambda { [n] n == 1 ? :one :
95 n == 2 ? :two : :other } } },
96 :th => { :i18n => { :plural => { :keys => [:other], :rule => lambda { [n] :other } } },
97 :tr => { :i18n => { :plural => { :keys => [:one, :other], :rule => lambda { [n] n == 1 ? :one : :other
98 } } }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
```

A continuación muestro una pantalla de como luce Discourse en su sección de bugs en general (fig 3.6), fue en esta sección donde buscaba bugs que pudiera resolver para poder contribuir a esta comunidad de código abierto, era un poco difícil ya que al ser la aplicación del co-fundador de Stack Overflow, tiene mucho auge y mucho tráfico de usuarios, por lo que los bugs eran resueltos en menos de una hora, máximo un día generalmente y esto resultaba en dificultad para poder elegir un bug y después solucionarlo.

Figura 3.6 Discourse en la sección de bugs.

bug ▾	Latest	New (8)	Unread	Top	+ New Topic	O ▾
Topic	Users	Replies	Views	Activity		
<p>📌 Category definition for bug</p> <p>A bug report means something is broken, preventing normal/typical use of Discourse. Do be sure to search prior to submitting bugs. Include repro steps, and only describe one bug per topic please.</p>		18	1.9k	Feb '13		
<p>Blockquotes On Profiles • new</p>		0	30	5h		
<p>🔒 After upgrade, some badges came with no descriptions • new</p>		5	43	5h		
<p>Newly approved members remain visible after approval</p>		6	463	8h		
<p>The pages for the tags with the «.json» suffix are rendered incorrectly</p>		1	74	9h		
<p>Viewing post from "group" Topics or Posts lists shows `0` (zero) voters • new</p> <p>poll</p>		1	76	9h		
<p>`j` + `k` keyboard shortcuts and small windows • new</p>		1	77	9h		
<p>List of contents that are not displaying custom text • new</p>		0	46	23h		

RESULTADOS OBTENIDOS

El proyecto aún se encuentra en marcha se terminará en junio de este año, se esperan muchos más resultados dentro de estos dos meses por venir, pero de momento puedo dar algunos obtenidos hasta este instante.

El más grande resultado obtenido desde mi punto de vista es el aprender a convivir con un equipo de trabajo y aprender a solucionar los problemas en conjunto, es muy difícil, de hecho me atrevo a decir que no existen equipos de desarrollo en los que una persona haga todo el trabajo a nivel profesional, es por esto que es una parte demasiado importante aprender a trabajar en conjunto, para esto fue muy importante el uso de las metodologías ágiles de desarrollo, el realizar pair programmings, daily meetings, TDD, entre otros, ayudan a los equipos a enfocarse mejor en lo que tienen que hacer y hacerlo de una mejor manera.

Otro resultado obtenido importante para mí fue el poder trabajar en conjunto con grandes desarrolladores dentro de la plataforma de Discourse, los mantenedores no solo te ayudan en casi todo lo que preguntes si no que te alientan a que sigas adelante y tratan de siempre marcar un camino por el cual deberías de seguir sin darte la respuesta, buscan que tu aprendas por ti mismo y eso es muy invaluable.

Se espera que para los finales de junio todos los miembros de Nearsoft Academy estén trabajando para un cliente, de no ser así serían contratados por Nearsoft o se daría referencia de ellos para otras empresas del mismo giro.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Al culminar mi estancia profesional comprendí que nunca se para de aprender, siempre habrá personas que sepan mucho más que tú, es importante tratar de trabajar en conjunto con esas personas ya que de ellas aprenderás mucho, también es importante recalcar que cada empresa tiene su propia cultura y al menos en Nearsoft se tiene esa cultura de ayudar a quien lo necesita, siempre y cuando esa persona que necesita ayuda alze la voz pidiendola.

Dentro de estos tres meses se aprendieron demasiadas cosas, nuevos lenguajes, nuevas metodologías, nuevas herramientas, por mencionar algunas, yo creo que todo esto no habría sido posible sin la ayuda de todos los miembros de mi equipo, ya que no hubo alguna tarea que nos fuera encargada que no la resolviéramos como un equipo y no de forma individual, esto es básicamente la estrategia del proyecto, borrar ese ego que muchos desarrolladores tenemos de decir que podemos hacer las cosas por nosotros mismos y tratar de resolverlas en conjunto con nuestro equipo, ya que muy posiblemente trabajes dentro de un equipo de desarrollo en un futuro y de esta manera puedes aprender mucho más de lo que aprenderías por ti mismo.

Recomendaciones

Las recomendaciones que daría a futuros practicantes es que traten de encontrar un lugar en donde puedan desarrollar sus habilidades, independientemente de su magnitud, un lugar donde se encuentren cómodos y puedan trabajar con gusto y no lo vean como una obligación, pienso que eso es lo mejor que te pueda pasar, ver tu trabajo como un hobby en vez de verlo como una obligación y eso solo puede darse en un trabajo con una cultura que tu disfrutes.

RETROALIMENTACIÓN

Fortalezas

Al terminar mi estancia profesional pude darme cuenta que una de mis fortalezas fue la capacidad para aprender de los demás, es decir resolver problemas de distintas magnitudes, además de entablar relaciones con personas dentro de la empresa que en un futuro pueden ser vitales.

Debilidades

Durante el proceso de resolución de problemas de Google code jam utilicé demasiadas malas prácticas de programación, ya que no tenía conocimiento previo de los lenguajes que utilicé, esto también se debió al tiempo, ya que solo duró una semana este ciclo de problemas, en un futuro planeo reescribir el código que hice mejorandolo y usando buenas prácticas.

Oportunidades

El trabajar en la aplicación de Discourse como un mantenedor fue algo que me gustó demasiado, tanto que de vez en cuando aún sigo manteniendo contacto con los que trabajaron conmigo dentro del proyecto, ayudando en lo que pueda cuando tengo tiempo, tuvo mucho que ver que el stack del framework que utiliza la aplicación es Ruby, un lenguaje que disfruto demasiado, en el futuro planeo seguir resolviendo bugs ya no por obligación del programa si no como reto personal para ayudarme a formarme más profesionalmente hablando.

Recomendaciones al programa de ISI

Después de haber realizado mis prácticas profesionales me di cuenta, que al menos para el área de desarrollo de software es sumamente importante que el contenido de las materias de programación sea actualizado. Por ejemplo en la optativa de sistemas basados en Web lo ideal sería que ya no solo se viera HTML y algún lenguaje de Back-End sino abarcar también MVC ya que ninguna empresa potencia utiliza algo que no sea un framework que utilice esto. Otra recomendación sería quitar alguna de las materias que están hasta el octavo o noveno semestre enfocadas más al área de industrial e introducir materias con contenido a colaboración en equipo, metodologías ágiles de desarrollo y tal vez crear una materia dedicada solamente a SCRUM. Yo sé que es bueno tener conocimiento acerca de los procesos industriales porque tal vez nos toque trabajar en una empresa o industria, sin embargo si no tenemos bien las bases de cómo trabajar dentro de un equipo de desarrollo todas las generaciones saldrán sin siquiera saber o tener una idea de cómo se trabaja en una empresa real, pienso que los futuros alumnos lo agradecerían mucho, otra recomendación sería que los maestros utilizaran Git para revisión de proyectos y que le dieran seguimiento a los commits que se realicen en los repositorios, de esta manera estarían obligando a los alumnos a utilizar un software de revisión de versiones y tendrían una bitácora de los cambios y sobre lo que han estado trabajando, esto para las materias como Programación Visual y en Internet, Tópicos avanzados de programación, Programación 2, entre otras.

REFERENCIAS

- [1] <https://atom.io/>
- [2] <https://www.jetbrains.com/idea/>
- [3] https://java.com/en/download/faq/whatis_java.xml
- [4] <http://php.net/manual/es/intro-whatism.php>
- [5] <http://www.rubyonrails.org/es/>
- [6] <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [7] <https://msdn.microsoft.com/es-mx/library/dd233154.aspx>
- [8] <https://golang.org/doc/>
- [9] <http://elixir-lang.org/>
- [10] <http://docs.spring.io/spring/docs/current/spring-framework-reference/pdf/spring-framework-reference.pdf>
- [11] <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- [12] <https://git-scm.com/book/es/v1>
- [13] <http://books.sonatype.com/mvnref-book/pdf/mvnref-pdf.pdf>
- [14] <https://it-ebooks24.com/ebook/creating-development-environments-with-vagrant>
- [15] https://www.docker.com/sites/default/files/WP_Intro%20to%20container%20security_03.20.2015%20%281%29.pdf
- [16] <https://www.virtualbox.org/>

[17] <http://www.itworld.com/article/2832994/open-source-tools/discourse---a-new-type-of-forum.html>

[18] <http://helpy.io/?source=ghh>

[19] <https://travis-ci.org/>

[20] <http://resonate.duarte.com/#!/page0>