

Reporte de conferencias Tékhne 2018

Antonio Emiko Ochoa Adame

28 de octubre de 2018

1. Conferencia 1: “Digital Transformation and Software/Requirements Engineering”

Esta conferencia trató el tema de la transformación digital, lo que esta implica y las ventajas que trae consigo. Una empresa debe que tener una transformación digital si quiere seguir compitiendo con las demás empresas que están allá afuera, las cuales son potencias en el sector.

Sea de software o no, una empresa necesita llevar a cabo esta transformación ya que debe adaptarse al mercado tanto actual como futuro; en este caso, la tecnología a través de las web apps, aplicaciones móviles y otros medios similares está dominando el mercado, por lo tanto, es necesaria esta adaptación si lo que se desea es mantenerse en el juego por mucho más tiempo.

Como ejemplo se puede mencionar el caso de Netflix vs Blockbuster. Netflix empezó a buscar nuevas formas de introducirse en el mercado; una de ellas fue la renta de películas por correo. Esto no está directamente relacionado con la transformación digital, o por lo menos eso parece. Netflix, en su búsqueda por estas nuevas estrategias llegó antes que Blockbuster a la transformación digital donde pudo anticiparse al “futuro” para así poder superar a su competidor, el cual actualmente se considera prácticamente extinto.

Otro ejemplo de lo anterior es Uber. Esta empresa logró introducirse en un mercado difícil, dominado prácticamente por los taxis y sus sindicatos. El gran éxito de dicha empresa se debe (entre otras cosas como el servicio y su trato hacia el cliente) al hecho de que el núcleo de la empresa está basado en software, ya que a través de este se realizan las transacciones y contacto, por ejemplo.

2. Conferencia 2: “Green Route, la aplicación para una movilidad inteligente”

Green Route es una aplicación hecha gracias al fware y la plática se desenvuelve en base a este último. Los datos son muy importantes en la actualidad, por lo tanto, se está promoviendo una forma de obtener datos de manera gratuita y libre. Estos datos pueden ser usados por las empresas para crear aplicaciones y servicios basados en estos datos que, además, están respaldados por instituciones

importantes como el gobierno u otras compañías. Se explicó el fiware como una forma forma de API o arquitectura de consulta libre al cual los desarrolladores e ingenieros tiene acceso con el fin de utilizar los datos proporcionados por dicha plataforma para realizar aplicaciones y servicios y satisfacer las necesidades de sus clientes.

Se habló también sobre SmartSDK que es el medio físico necesario para obtener los datos que esta empresa y otras proporcionan. A través de este SDK se puede tener acceso a varios servicios y datos de instituciones como Infotec, Ubiwhere, IHOP, entre otras.

Infotec promueve el uso de estos servicios a través de módulos los cuales se denominan "modelos de datos" que pueden dar información desde tráfico (como Google Maps o Waze), sectores lluviosos de la ciudad, lugares con polen, hasta servicios como cloudino que es una forma fácil de poder conectar un arduino con la nube para poder controlar remotamente; esto es muy útil ya que soluciona el problema de reiniciar el dispositivo manualmente yendo al lugar donde está instalado ya que simplemente con un comando a distancia se le pueden dar órdenes y el dispositivo seguirá funcionando de igual manera.

3. Conferencia 3: “Machine Learning”

Esta conferencia fue dada por un doctor de la Universidad Michoacana de San Nicolás de Hidalgo el cual comentó las diferencias entre la Inteligencia Artificial (IA), Machine Learning, Deep Learning y su relación entre ellas.

Explicó que hay varias formas de “Inteligencia Artificial” y que dependiendo del área de trabajo te puede interesar una u otra ya que tienen diferentes aplicaciones. A estos diferentes tipos de inteligencia que se puede obtener por una máquina se le clasificó en 4 sectores donde 2 de estos hacen referencia a la inteligencia racional y la inteligencia humana. El caso de la intersección de inteligencia humana es despreciada ya que en el campo de la ciencias informáticas no supone una ventaja ya que lo que realmente se intenta obtener es una inteligencia parecida a la nuestra combinada con el potencial y velocidad de una máquina. Por otra parte está la inteligencia racional donde sí que se comparte la forma humana racional de, por ejemplo, pensar o la forma en que se asocian ideas para poder formar una idea más grande.

El área en la que trabaja el doctor Juan José es en la inteligencia racional humana en interacción con aquella que responde a un estímulo. Parece un poco complejo, pero no lo es; simplemente es una inteligencia artificial (que puede ser un robot o simplemente software) que interactúa con el medio en el que se encuentra. De esta forma se recibe un estímulo del medio, se procesa la información percibida y se obtiene una respuesta a dicho estímulo que puede ser una acción o un proceso similar.

4. Conferencia 4: “Importancia de la Optimización en la eficiencia de Algoritmos”

Juan Barrera habla sobre la gran importancia de la mejora de los algoritmos en cuanto a eficiencia se refiere. Empezó dando algunos ejemplos sobre Big O notation con el fin de que se pudiese desarrollar el tema de tal forma que todos entendieran la complejidad de los algoritmos y su implicación. Uno de los ejemplos fue $O(n)$ donde la complejidad es lineal. Un ejemplo de esto puede ser un ciclo for donde n es el número de veces que se va a ejecutar un ciclo o recorrer un arreglo o una lista donde n sería el número de elementos que tiene dicho arreglo o lista. Un ejemplo de complejidad logarítmica $O(\log n)$ es la búsqueda binaria donde por cada elemento descartado la búsqueda del elemento se reduce a la mitad. Un ejemplo de complejidad constante $O(1)$ puede ser una búsqueda en un diccionario en Python o una tabla de Hashes cualquiera.

La complejidad algorítmica indica cuánto tiempo o coste computacional puede requerir un proceso para lograr su objetivo. Siempre se tiene un "peor caso", un caso ideal y un caso común, donde este último suele ser el de referencia normalmente. Si se toma de ejemplo el caso de buscar un elemento de un arreglo con un algoritmo de complejidad $O(n)$ el mejor caso sería que el elemento se encontrara en la primera posición, el peor caso sería que se encontrara en la última posición y el más común es que se encuentre en cualquier otro lado (no solamente en medio). En el caso de hacer la misma operación (buscar un elemento en un arreglo) pero en este caso utilizando búsqueda binaria los casos son los siguientes: el mejor caso sería que el elemento estuviera justo en medio, el peor caso podría ser cerca del medio y el caso común en los lugares restantes.

Entonces, ¿Porqué es importante la complejidad de los algoritmos y su optimización?. Tomando el ejemplo anterior donde se busca un dato en un arreglo y considerando empresas como Google donde se trabaja con millones o billones de datos de usuarios distintos, usar un algoritmo lineal tomaría demasiado tiempo para hacer una búsqueda de los datos de un solo usuario. Esto no es nada eficiente considerando la cantidad de usuarios y tiempo de respuesta de un servidor que se puede tener, por lo tanto, es un desperdicio inmenso de recursos, de tiempo y que esto a su vez implica pérdida de dinero.

Empresas grandes como Google, Facebook, Apple, Amazon, Microsoft, etc., buscan personas que sean capaces de desarrollar código de manera optimizada y que cumpla con las necesidades de la empresa. Esto está involucrado con el hecho de que en este tipo de empresas no es fácil conseguir trabajo ya que muchos egresados no cumplen con los requisitos mencionados anteriormente, incluso si estos son buenos con ciertos lenguajes de programación o tienen habilidades diferentes.

Casi siempre se puede optimizar un código.