

# LECTURAS

Lecturas 1, 2 ,3, 4 y 5

## Descripción breve

El documento contiene pequeñas reseñas de las lecturas subidas a Trello

Tristán Nathaniel

# Contenido

Lectura 1 .....	2
Las 11 razones por las que el desarrollo de software profesional nunca dejará de existir .....	2
Lectura 2 .....	2
Importancia del Software .....	2
Lectura 3 .....	3
Leyes Famosas De Desarrollo De Software.....	3
Lectura 4 .....	3
50 años de la Ingeniería de Software Problemas, Logros, Tendencias y Retos ..	3
Lectura 5 .....	4
Metodologías tradicionales vs. metodologías ágiles .....	4

## **Lectura 1**

### **Las 11 razones por las que el desarrollo de software profesional nunca dejará de existir**

La lectura habla de las razones las cuales los programadores siguen y seguirán existiendo, una de estas razones es la de que los software siempre avanzan, siempre se actualizan y por ello siempre abra gente que necesite adaptar sus cosas no muy actualizadas a lo mas actualizado eso es un punto real ya que por ejemplo una persona que compra un teléfono hoy en prácticamente 1 año ese teléfono se empezara a hacer obsoleto por los avances que el software tiene y en esto pone el ejemplo del ordenador de orientación del Apollo que comparado con un teléfono celular de hoy en día es un ordenador obsoleto ya que procesa muy poco comparado con cualquier teléfono celular de hoyen día, además de los rápidos avances en el software este toca otro punto que es verdad, que dice que las maquinas ya están siendo parte del centro de nuestras vidas como por ejemplo puedes comprar un boleto de avión desde la comodidad de tu casa y eso por ejemplo es algo que no se podía hacer en el pasado, otro ejemplo que pone es el de localizar donde se encuentra un taxi para poder usarlo o pedir uno que vaya a tu ubicación actual esto ya es algo que con solo tu teléfono puedes hacer hoy en día, otro punto que toca es el de buscar gente buena en desarrollo de software esto es porque en un punto de estos avances abra mucha gente que sepa programar pero no abra tanta que sepa programar cosas complejas ya que necesitan la experiencia y esto ara que las solicitudes laborales no bajen porque todos sepan programar, por esto y los otros puntos que toca la lectura opino lo mismo, que el desarrollo de software le queda muchísimo tiempo de vida que hasta incluso algunos proyecto software antiguos van a durar eternamente.

## **Lectura 2**

### **Importancia del Software**

Esta lectura trata de la importancia de la calidad del software y esta involucra demasiado a los profesionistas y profesores de las carreras que involucran a esta, ya que sin estos no tienen la educación adecuada del software o si estos profesionales y profesores educadores no son suficientes o no abarcan las metas para la buena calidad del software las nuevas generaciones crecerán y saldrán al mundo con un concepto erróneo de lo que es el desarrollo y calidad del software, y de hay es tan importante por qué los encargados de manejar carreras inviertan los recursos suficientes para tener un buen desarrollo como profesionistas dentro del área y tengan en cuenta que si los profesores no alcanzan las metas del aprendizaje en el desarrollo del software el software ira yendo a peor en el traspaso del tiempo, pero si este es invertido correctamente los errores en software prácticamente no aparecerán, y esto lo podemos ver en el pasado, ya que los llamados pantallazos azules hoy en día no son muy usuales de ver en softwares con una buen

inversión por ejemplo los pantallazos de Windows prácticamente en las versiones mas actuales no aparecen pero en cambio, en las versiones mas “viejas” aparecen muy frecuentemente y esto pasaba por que las áreas de manejo de software prácticamente eran muy pocas o muy específicas a un área pero esto con el paso de los años y muchos avances en el desarrollo y calidad de software han ido disminuyendo drásticamente; y por esto es tan importante el buen desarrollo de softwares de calidad y la correcta inversión en el desarrollo de profesionistas de esta área.

### **Lectura 3**

#### **Leyes Famosas De Desarrollo De Software**

Esta lectura te cuenta las “leyes” mas pasadas o que mas ocurren durante la creación o desarrollo del software estas leyes fueron creadas con un trasfondo lleno de experiencias como por ejemplo la “ley de noventa y noventa” que te cuenta que al momento de hacer programación o mejor dicho desarrollar un software este empieza “fácil y rápido” asta que llegas al figurativo noventa porciento y te quedas estancado pensando la solución para hacer lo que necesitas, es decir explicarle a una maquina como hacer lo que necesitas de la manera mas clara, simple y corta posible, primeo clara porque lo tiene que hacer como lo dices y quieres después simple porque si este es muy complejo tardara mas en ejecutarse o será mas pesado y por ultimo de la mano con la anterior simple que es las anteriores juntas; también habla de las malas ideas de en pocas palabras invertir en personal para la creación y desarrollo de software cuando el software va por la mitad, esta es una mala idea porque si contratas gente para que “te ayude a acabar mas rápido” pasara exactamente lo contrario porque cuando tu tardaste por ejemplo tres meses en llegar a donde esta tu desarrollo de tu programa ellos tardaran como mínimo lo mismo que tu para comprender todo lo que llevas y eso no es prácticamente todo lo contrario a lo que buscabas desde un principio que era tardar más, además entre esas leyes más famosas del software existe una de ellas que no solo aplica al software la cual es la ley de Murphy y dice que si algo puede salir mal lo ara, por desgracia esta es una de las leyes que más aparecen en nuestras vidas cotidianas y por supuesto durante el desarrollo de software; por esto que he comentado de algunas leyes del desarrollo de software y otras que no, concluyo que estas leyes son muy útiles a la hora de pensar desarrollar un software tenerlas y tomarlas en cuenta al momento de iniciar ya que estas reúnen experiencias de otros desarrolladores y las comparten para facilitarnos las cosas.

### **Lectura 4**

#### **50 años de la Ingeniería de Software Problemas, Logros, Tendencias y Retos**

La lectura habla de como su titulo dice algunos problemas, logros y retos que la ingeniería en software lleva tratando por alrededor de cincuenta años, por ejemplo habla de algunos problemas que se mencionan también en lecturas pasadas como en la importancia del software o las leyes más famosas de estas como la calidad del software en aquellos que son mas integrados en la sociedad, como aquellas que dependen de una o muchas vidas, otro problema mencionados son los costos para la preparación de un buen software como por ejemplo en la preparación de IBM que se gastaron 50 millones de dólares al año durante su preparación, y una inversión de al menos 5,000 años-hombre, es decir los años sumados por persona durante la preparación por ejemplo 5,000 mil personas por un año pero no fue el caso ya que probablemente se fue su vida en ello para algunas personas, también se mencionan los problemas de gestión que seria el no cumplimiento en las fechas de entrega y sus especificaciones requeridas, cambiando a las tendencias toca los puntos de que hoy en día es mas barato empezar software en la nube y facilita su manejo remoto haciendo que

no tengas que tener un disco duro en cada lado que necesites usarlo, y por ultimo los retos que tiene el software para el futuro son por ejemplo el cambio profundo y radical en los métodos y técnicas utilizados para concebir, diseñar, desarrollar, probar y desplegar software; por esto creo que la historia es importante para ver de donde vienen las cosas y así ver los rumbos que pueden tomar con una visión muy clara de las metas y retos que se desean cumplir.

## **Lectura 5**

### **Metodologías tradicionales vs. metodologías ágiles**

Esta ultima lectura abarca el campo mas importante que es el desarrollo de software y sus opciones para la planeación, actividades y etapas para la creación de esta, además nos habla de las diferencias, ventajas y desventajas de las metodologías tradicionales y las metodologías ágiles, toca un punto muy interesante ya que comenta que lo mejor es tener experiencia en la metodología tradicional ya que esta es prácticamente la base para seguir las metodologías ágiles y sin experiencia de esta las metodologías ágiles no serían tan buenas como las tradicionales en este caso, pero en el caso de que tengas una buena experiencia con los métodos tradicionales los métodos ágiles son mas compactos.