

IS Retrospectiva

Al mostrar interés por primera vez en la carrera de Ingeniería de Software impartida en la facultad de matemáticas, para mí, el propio concepto de un ingeniero de software recaía principalmente en un enfoque a la programación y creación de software, como si el hecho de ser uno significara tener que ocupar un puesto exclusiva y únicamente de programador, cosa que no podría estar más alejada de la realidad.

Con lo que he aprendido a lo largo de estos 3 meses dentro de la carrera mi concepto cambió drásticamente hasta el punto de ver de las principales características que debe tener un ingeniero de software, más que el dominio de lenguajes de programación y *problem solving*, son las habilidades comunicativas y la correcta administración de proyectos de software: desde la comunicación con el cliente, hasta el tener en consideración el tiempo, personal y recursos que tienes a tu disposición, puesto que prácticamente siempre se estará trabajando junto con otras personas y con múltiples limitantes de las que depende la integridad del proyecto, y si no existe la comunicación o los métodos correctos para manejarlo se verá comprometida la eficiencia o incluso la propia realización del mismo.

Estas características son llamadas *soft skills*; competencias, atributos sociales y personales necesarias para un desarrollo óptimo en el ámbito profesional, formando parte del kit de competencias de un ingeniero de software junto con su conocimiento y habilidades en el propio desarrollo de software.

IS vs Otros Roles

Existen múltiples áreas de desarrollo de software, entre las cuales actualmente hay múltiples roles especializados, como *back-end*, *front-end*, *fullstack*, *DevOps* o *seguridad*, pero hoy me enfocaré específicamente en el rol de *fullstack*, término el cual muchos lo consideran controversial y e incluso que no se encuentra muy bien definido, aun así yo no lo consideraría un rol estrictamente especializado, puesto que dentro de lo que cabe se basa en la comprensión y manejo competente tanto del *back-end* como del *front-end*, abarcando ambos de los componentes esenciales que conforman una aplicación web. Aun así es un rol que requiere de muchos conocimientos y dominio de múltiples tecnologías y lenguajes de programación.

Si tomamos el rol de un ingeniero de software como punto de referencia para compararlo con la definición anterior de un desarrollador *full-stack*, este tiene una gama más amplia de roles que puede llegar a abarcar además del desarrollo web. El desarrollador tiene un amplio conocimiento de lenguajes de programación y es muy bueno dentro de su área, pero

desconoce las demás y se encuentra limitado fuera de la que domina. El ingeniero puede estar presente no solo en el desarrollo, sino también estar presente en todas las etapas del proyecto incluyendo la gestión de los recursos y del personal, y comunicándose con las demás partes del proyecto como haría un DevOps, otorgándole con mucha mayor flexibilidad que otros roles.

Tema libre U1

Artículo “Pandemic programming: How COVID-19 affects software developers and how their organizations can help”

La pandemia que se vivió en 2020 a raíz del virus Covid-19 trajo un largo periodo de distanciamiento social, en el que, por una parte, hubieron personas perdieron sus empleos y muchos negocios se vieron afectados o directamente cerraron, y por otra parte hubieron quienes tuvieron la oportunidad de trabajar desde su hogar según dependiendo del trabajo que tuvieran. En este caso me enfocaré en los ingenieros de software, los cuales caen dentro de la segunda categoría e incluso se podría decir que se vieron beneficiados por este suceso por más mal que suene, y aunque esta transición es algo que es posible dentro de su campo de trabajo, se le podría decir que ocurrió de manera drástica y no precisamente tranquila para todos.

Antes de la pandemia ya había muchos ingenieros de software que estaban trabajando desde casa y estaban acostumbrados a esta dinámica, sin embargo, había muchos que no estaban en esas condiciones y tuvieron que vivir el cómo su espacio laboral y su hogar convergieron, y aunque uno pensaría que el impacto en la eficiencia sería mínimo, en realidad no fue así; el artículo que hoy cité demuestra que la pandemia realmente afectó la estabilidad emocional de muchos los ingenieros de software que se vieron afectados por ansiedad, insomnio, aislamiento, distracciones, incertidumbre, y una gran cantidad de estrés, lo cual impactó directamente su bienestar y eficiencia en sus actividades laborales.

Por esto mismo es muy importante que al hacer el cambio no se asuma que los empleados tienen los recursos o están en la mejor situación para hacer un cambio así de abrupto. Con esto quiero dar a entender que claramente no había otra opción, pero esto no es una excusa y se debería poner mayor importancia en el bienestar de los ingenieros de software que trabajan de forma remota para asegurarse de que tienen lo necesario para trabajar en esa posición y que se encuentran en el estado emocional adecuado para cumplir su trabajo, ya sea abriendo la posibilidad de recibir apoyo psicológica y otorgando materiales y recursos como podría ser una silla ergonómica, un equipo de cómputo, pagar un porcentaje del gasto del internet, por mencionar algunos.

Esto muestra que, por más que se pudiera decir que el perfil de un ingeniero en software no tendría problema alguno trabajando desde casa, en realidad no es así y depende de las circunstancias tanto del contexto mundial como personal de cada uno y se debe tomar con la

seriedad que se merece, así como asegurarse de compensarlo en la manera de lo posible si se está en el lugar del empleador o empresa para beneficio de ambas partes.

Empirical Software Engineering. 2020. Pandemic programming. Recuperado el 15 de Octubre de 2022 de:

https://www.researchgate.net/publication/344342621_Pandemic_programming_How_COVID_19_affects_software_developers_and_how_their_organizations_can_help

Reflexión x2

Como forma de evaluación para FIS, se estableció una actividad que se basa en reflexionar sobre distintos temas de relevancia relacionados a ingeniería de software, pero ¿Cómo se asegura que se hace una reflexión correctamente? ¿Qué es lo que compone a la misma?

La reflexión se podría definir como un proceso que surge del pensamiento crítico y detenido sobre un tema en especial cuya argumentación está basada en hechos objetivos.

Si tomamos como referencia *IS vs Otros Roles*, se establece la definición más aproximada de lo que es un desarrollador *full-stack* y el alcance que tiene de acuerdo a sus conocimientos y su posición, y luego se hace una comparación de este con las competencias de un ingeniero de software como lo son las habilidades sociales y administración de proyectos y recursos por poner un ejemplo, terminando con una conclusión que se basa en estos datos mostrando la diferencia entre ambos, manteniendo la coherencia y objetividad en todo momento.