

Cinco consejos de Creditas para mejorar destrezas de programación y ser mejor desarrollador

Los perfiles tech se enfrentan, en ocasiones, a retos a la hora de trasladar su día a día a los equipos que no están tan familiarizados con la tecnología. Creditas comparte consejos que pueden ayudar a que los programadores desempeñen mejor sus tareas

[Creditas](#), la plataforma líder en América Latina de soluciones de consumo y préstamos 100% online, confía en su oficina en Valencia como centro tecnológico y hub de innovación. El equipo en España está, en su mayoría, formado por perfiles *tech* que, en ocasiones, se enfrentan al reto de transmitir y compartir sus tareas y su día a día con equipos menos orientados al trabajo puramente tecnológico.

“Ser buen desarrollador no es solo ser el mejor a la hora de diseñar y crear un producto. También es importante ser bueno comunicándolo y compartiéndolo con el equipo. Muchos nos ven como perfiles más solitarios o independientes, pero formamos parte de equipos que necesitan y quieren estar al día de lo que hacemos. Poner en común con el resto de compañeros nuestro día a día es muy positivo para todos. En ocasiones supone un reto, sí, pero merece la pena enfrentarse a él y contar con consejos para superarlo”, explica Tiago Fernandes, Senior Software Engineer en Creditas.

Los expertos del hub de innovación de [Creditas](#) ofrecen cinco consejos que pueden ayudar a mejorar las destrezas de programación, el trabajo en equipo y la eficiencia de los productos desarrollados:

Apostar por una buena comunicación suele ser la mejor solución

Es evidente que en una compañía no se trabaja de forma aislada. La interacción con otros equipos y miembros de la empresa es inevitable y la comunicación se ha convertido en un factor clave y constante de la rutina del trabajador. Es importante tener en mente distintos factores que pueden favorecer que la comunicación sea lo más fluida y eficiente posible.

Es importante comunicar con empatía. A la hora de entender y estudiar un producto desarrollado por el equipo de programación, se debe tener en cuenta que no todo el mundo cuenta con un perfil con conocimientos de tecnología. Es importante escuchar con calma y esforzarse por comprender los distintos puntos de vista, asegurando que se hacen todas las preguntas necesarias para conseguirlo. Además, en la mayoría de las ocasiones, crear un entorno menos formal y relajado con cafés, almuerzos y reuniones informales de equipo puede ayudar a conocer un poco más a la gente sin la presión del trabajo y puede ser

clave a la hora de preparar el terreno para reuniones profesionales y proyectos en común.

También se debe medir el uso de la jerga técnica. El público al que se dirige el mensaje es clave y se debe intentar explicar con la cantidad de detalles adecuada para cada perfil. Es importante desglosar los detalles técnicos en versiones simplificadas o en analogías que puedan ayudar a que la comunicación fluya y a que los conocimientos sean transmitidos y comprendidos claramente por todos los participantes.

Infórmate, mantente al día y lee todo lo que puedas

Antes de empezar a utilizar cualquier herramienta de terceros, es esencial leer toda la documentación disponible. Se suele pensar que leer manuales es contraproducente porque retrasa el empezar a codificar, pero conocer la API, su arquitectura y cómo utilizar correctamente cada herramienta, reducirá los problemas y los atascos. Además, al empezar a trabajar con un nuevo lenguaje o biblioteca es recomendable utilizar los repositorios públicos online (Github, Gitlab, etc.) como recursos de aprendizaje. Una buena regla general es seguir proyectos conocidos y tratar de leer y entender su código base. Documentarse es clave, pero inspirarse todo lo posible en el trabajo de los demás, también.

Céntrate en el producto

El valor de un ingeniero también se mide por su contribución activa en el proceso de diseño y creación de un producto. Las compañías valoran que sean capaces de aportar nuevas perspectivas, ideas y sugerencias al gestor de productos, de explicar las ventajas y desventajas de estas y de ayudar al gestor de productos a elegir la mejor opción en el momento y lugar concretos y, para esto, es importante poder construir las características de cada producto en torno a entregables verticales significativos. Puede ser interesante desglosarlas y dividir las para enfrentarse a tareas más pequeñas con las que se aporte valor de principio a fin antes de que el producto se lleve a producción. Con esta estrategia se reducirá la cantidad de trabajo a una cantidad asequible y será más fácil desarrollarla, comprenderla, comunicarla, probarla y aumentar el valor añadido del producto.

No obsesionarse con reinventar la rueda

La falta de una característica concreta de las soluciones necesarias al desarrollar un producto puede hacer que se caiga en el "construyo mi propia versión". Sin embargo, si hay una característica necesaria que no está presente, lo mejor es intentar pedirla o contribuir a sus proyectos.

Además, es importante centrar el desarrollo de un producto en las características de este, no en los bloques de construcción. En la mayoría de las ocasiones, utilizar las bibliotecas de software es una buena decisión y, si se puede externalizar su mantenimiento, se podrá disponer de más tiempo libre para centrarse y aportar valor al proyecto.

Desarrolla y trabaja con el objetivo final en mente

Independientemente de si se aplica TDD o no, se deben centrar los esfuerzos en que la característica o tarea se complete de principio a fin y, para ello, es

esencial preparar una batería de pruebas previas. Según los expertos de [Credits](#), nunca está de más decir que las pruebas son una parte fundamental del trabajo del desarrollador ya que, además, pueden ayudar a moverse más rápido y con confianza, entregando características libres de errores. Desde las pruebas unitarias, hasta las de integración, todas pueden marcar la diferencia entre una experiencia de codificación indolora y divertida y una experiencia estresante.
