

GUION PARA VIDEO PD 2-3

-INTRODUCCIÓN -

[Insertar transición de sol saliendo debajo de las nubes] [Insertar soundtrack Morning Mood]

- **Narrador:** ¡Ey hola! Una disculpa, no los había visto detrás de mí. ¿Cómo dijeron? ¿Quieren aprender a desarrollar software? ¡Pues manos a la obra!
- **Intervención:** [Haciendo zoom a la persona que interviene] Ohh no, no de nuevo, ¡que alguien lo calle! [Insertar FX transición tv efecto de sonido para countryballs (sonido de impacto)] [Efecto de transición de tv]

-DESARROLLO -

- **Narrador:** [Mientras se acerca a la cámara] Primero tienes que saber cuáles son las habilidades que debe de tener todo buen ingeniero de software. Estas son:
[Ejemplificar cada uno usando los sims 4, la ejemplificación no debe durar más de 5 segundos, siendo el máximo a abarcar 40 segundos]
 1. Habilidades de aprendizaje.
 2. Conocimientos fundamentales, básicamente, saberle a la computadora y matemáticas.
 4. Gestión de proyectos
 5. Habilidades interpersonales o softskills
 6. Adaptabilidad y flexibilidad ante los cambios
 7. Entre otros[Corte abrupto a la cámara de Vr chat, el personaje la está viendo fijamente]
- **Narrador:** Bueno, siguiendo con el recorrido, hay una fase muy importante en el desarrollo de software, ¿adivinas cuál es? ... [Dejar silencio de 2 segundos]
¡Correcto! Es la fase de requerimientos. En la cual hablaremos de las habilidades que un ingeniero de software necesita para crear requerimientos la cuales son:
 - Análisis de información
 - Diseño de prototipos de UI
 - Escritura en lenguaje natural
- **Narrador:** [Transición a los sims 4 para ejemplificar con voz en off] Por ejemplo, en nuestro proyecto de desarrollo el lenguaje natural fue muy importante para escribir las especificaciones de casos de uso, y diseñar los prototipos.
- **Narrador:** Pero ¿cómo le hacemos para realizar los requerimientos? ... [Dejar silencio de 2 segundos] [Insertar sonido de idea cartoon] ¡Ya se! Podemos usar metodologías ágiles. [Mirando al horizonte] Las metodologías ágiles son alternativas a los modelos de ciclo de vida habituales, estas se basan en una serie de principios de agilidad. [Voltea a la cámara] Nos centraremos en Scrum y XP o programación extrema.

- **Narrador:** En el caso de Scrum [**Colocarse al lado izquierdo y poner imagen de Scrum al lado derecho**] podemos ver los principios de “Entrega de valor”, “Mejora continua”, “ritmo de trabajo sostenible”, “Operativas visibles” y “personas sobre procesos”.
- **Narrador:** En el caso de XP o programación extrema [**Colocarse al lado derecho y poner imagen de XP al lado izquierdo**] podemos ver los principios de “Entrega de valor”, “Desarrollo iterativo e incremental”, “Atención continua a la excelencia” y “Operativa visible”.
- **Narrador:** [**Transición con efecto swish a Voz en Off usando Sims para ejemplificar**] En nuestro proyecto donde hay poco personal, ya que no hay *money*, y se cuenta con plazos de entrega cortos, se puede realizar ciertas adaptaciones para llevar a cabo estos principios. Se puede entregar valor al cliente y obtener una operativa visible realizando reuniones diarias o casi diarias e invitando a nuestro querido Product Owner, se observa un desarrollo iterativo se observa por medio de entregas pequeñas llamadas *sprints* haciendo una retroalimentación del proceso de desarrollo al final de cada una se observa la mejora continua.
- **Narrador:** [**Transición a la cámara de Vr chat, el narrador esta de espaldas y se voltea**] Ya que estamos hablando sobre usar metodologías ágiles, ¿qué pasaría si los valores que se describen en su manifiesto, el cual es un documento que contiene valores y principios con nuevas formas de desarrollo de software causan conflicto en el desarrollo de nuestro proyecto? Es simple.
- **Narrador:** Básicamente hay que adaptar dichos principios a nuestro proyecto y no al revés. Por ejemplo, en el desarrollo de nuestro proyecto puede causar problemas obtener una retroalimentación constante, por ello se busca priorizar las personas sobre las herramientas y la retroalimentación por medio de reuniones... [**Dejar un silencio de 2 segundos y acercarse a la cámara**] se puede pensar en que haciendo esto se carecerá de un proceso claro para el desarrollo y que todo estaría en caos [**Ejemplificarlo con los Sims 4**], pero bastaría con adaptar las reuniones que dice ese principio a un proceso más estructurado, es decir, combinar una metodología ágil con aspectos de los ciclos de vida de desarrollo clásicos, encontrando un equilibrio que favorezca la organización, retroalimentación y mejora en el desarrollo del proyecto.
- **Narrador:** ¡Bien! Ya hablamos de lo necesario para hacer requerimientos y de lo efectivas que pueden ser las metodologías ágiles bien implementadas; en estas se usan historias de uso para determinar requerimientos, la pregunta es ¿Este método ágil en que se diferencia con las especificaciones de casos de usos de las otras metodologías?
[**Insertar efecto de transición simple**]
- **Narrador:** [**Ejemplificar utilizando los sims 4**] En la especificación de casos de uso es literalmente especificar qué es lo que va a hacer cada requerimiento y como el usuario debe de navegar cuando este usando el software, básicamente pura talacha. Por otro lado, las historias de usuario se hacen desde la perspectiva del usuario

donde se hacen las siguientes preguntas para describir mejor los requerimientos desde la perspectiva del usuario:

- **Narrador:** [Mostrar las preguntas en pantalla completa con voz en off de fondo] ¿Cómo usuario quiero? ¿Para qué lo quiero?
- **Narrador:** [Se hace la transición a VR chat] El uso de uno o de otro depende de la metodología usada en el proyecto. En nuestro caso se decidió usar las especificaciones de casos de uso por su alto nivel de detalle... [dejar 2 segundos para más suspenso] y porque yo les dije.
- **Narrador:** Lo interesante aquí es que del uso de algunos de estos métodos se puede empezar a crear prototipo de base de datos, interfaces, entre otros, en la etapa de diseño [corte y transición abrupta a personaje cerca de la cámara y le habla al espectador] shhhh, aquí entre nos esta etapa igual es muy importante en el desarrollo
- **Narrador:** En nuestro caso, el producto resultante fue el diseño de interfaces, en donde el método usado para crear este prototipo de interfaz fue usando los *Wireframes* de baja fidelidad, estos nos permitieron tener una idea básica de lo que será nuestras interfaces [El narrador se pone al lado izquierdo y se muestran imágenes del diseño de interfaces al costado derecho]
- **Narrador:** Creo que olvido algo... [Dejar un silencio de 2 segundos y hacer sonido de pensar] Bueno, si lo olvide no debe ser importante [Se corta abruptamente y se pone en pantalla un texto que diga PRUEBAS]
- **Narrador:** Auch, no hacía falta el golpe. En fin, perdónenme, las pruebas sí que son muy importantes, ya que nos ayudan a asegurar la calidad de nuestro software. Hay muchos métodos y ninguno es mejor que otro, todo depende del contexto de cada proyecto. Podemos resaltar dos métodos: [Ejemplificar ambos modelos usando los sims 4]
 1. **Modelos de desarrollo basados en pruebas (TDD).** En este se busca la recodificación constante y se realizan las pruebas durante casi todo el desarrollo, no solo en la codificación.
 2. **Modelo iterativo.** Se basa en entregas pequeñas, donde poco a poco se realizan modificaciones y mejoras. Las pruebas se suelen realizar en el diseño, codificación y después de esta de cada pequeña entrega.
- **Narrador:** [Cambiando a Vr chat] En nuestro proyecto usamos ambos métodos ya que en conjunto nos ayudan a que sea más eficiente el proceso de desarrollo.

-CIERRE-

- **Narrador:** ¡Ahora sí! Este es el final. Quiero aprovechar para mandar saludos a mi abuelita [Es interrumpido abruptamente]
- Intervención: Sí, sí. Ya es hora de terminar el video.
- **Narrador:** ¡Alto espera! No me cort- [Insertar créditos finales o solo una pantalla que diga FIN]

