**TECNOLÓGICO​ ​NACIONAL​ ​DE​ ​MÉXICO**

**INSTITUTO TECNOLÓGICO DE TIJUANA**

**SUBDIRECCIÓN ACADÉMICA**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

**SEMESTRE:** Agosto - Diciembre 2020

**CARRERA:**Ing. en sistemas computacionales

**MATERIA:**Minería de datos

**TÍTULO:​ ​**Practica 1

**UNIDAD​ ​A​ ​EVALUAR:​**​Unidad​ ​1

**NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:**QUINTERO NAVARRO ISRAEL 15210544

**NOMBRE DEL DOCENTE :**

Jose Christian Romero Hernandez

### **¿Qué es el pair coding?**

La programación por pares significa esencialmente que dos personas escriben código juntas en una máquina. Es una forma de trabajo muy colaborativa e implica mucha comunicación. Si bien un par de desarrolladores trabajan juntos en una tarea, no solo escriben código, también planifican y discuten su trabajo. Aclaran ideas en el camino, discuten enfoques y llegan a mejores soluciones.

**¿Cuales son alguno de los estilos de Programación por pares que existen ?**

#### Conductor y navegador

Estas definiciones clásicas de roles de programación de pares se pueden aplicar de una forma u otra a muchos de los enfoques de emparejamiento.

El **conductor** es la persona al volante, es decir, el teclado. Ella está enfocada en completar el pequeño objetivo que tiene entre manos, ignorando los problemas más importantes por el momento. Un conductor siempre debe hablar sobre lo que está haciendo mientras lo hace.

El **navegador** está en la posición de observador, mientras el conductor escribe. Ella revisa el código sobre la marcha, da instrucciones y comparte pensamientos. El navegador también está atento a los problemas más importantes, los errores y toma notas de los posibles próximos pasos u obstáculos.

La idea de esta división de roles es tener dos perspectivas diferentes sobre el código. Se supone que el pensamiento del conductor es más táctico, pensando en los detalles, las líneas de código a mano. El navegante puede pensar de manera más estratégica en su función de observación. Tienen el panorama general en mente.

#### 

#### Ping Pong

Esta técnica abarca el desarrollo basado en pruebas (TDD) y es perfecta cuando tiene una tarea claramente definida que se puede implementar de una manera basada en pruebas.

* "Ping": el desarrollador A escribe una prueba fallida
* "Pong": el desarrollador B escribe la implementación para que pase.
* El desarrollador B inicia el siguiente "ping", es decir, la siguiente prueba fallida.
* También se puede seguir cada "Pong" refactorizando el código en conjunto, antes de pasar a la siguiente prueba fallida. De esta manera se sigue el enfoque "Rojo - Verde - Refactorización": escriba una prueba fallida (rojo), haga que pase con los medios mínimos necesarios (verde) y luego refactoriza.

#### Strong-Style Pairing

Esta es una técnica particularmente útil para la transferencia de conocimiento, descrita con mucho más detalle por Llewellyn Falco aquí.

La regla: "Para que una idea pase de tu cabeza a la computadora, DEBE pasar por las manos de otra persona". En este estilo, el navegador suele ser la persona mucho más experimentada con la configuración o la tarea en cuestión, mientras que el conductor es un novato (con el idioma, la herramienta, el código base, ...). La persona experimentada permanece principalmente en el papel de navegador y guía al novato.

Un aspecto importante de esto es la idea de que el conductor confía totalmente en el navegador y debe sentirse "cómodo con una comprensión incompleta". Las preguntas sobre "por qué" y los desafíos a la solución deben discutirse después de la sesión de implementación. En un entorno donde una persona es un novato total, esto puede hacer que la combinación sea mucho más efectiva.

Si bien esta técnica roza la microgestión, puede ser una herramienta de incorporación útil para favorecer el "aprendizaje mediante la práctica" activo sobre el "aprendizaje mediante la observación" pasivo. Este estilo es ideal para la transferencia inicial de conocimientos, pero no debe usarse en exceso. Tenga en cuenta que el objetivo es poder cambiar de rol fácilmente después de un tiempo y salir del modo de microgestión. Eso será una señal de que la transferencia de conocimientos funcionó.

#### Pair Development

El "Pair Development" no es tanto una técnica específica para emparejar, sino más bien una forma de pensar sobre el emparejamiento. (Encontramos el término por primera vez en este hilo en la cuenta de Twitter de Sarah Mei). El desarrollo de una historia de usuario o una función generalmente requiere no sólo codificación, sino muchas otras tareas. Como pareja, son responsables de todas esas cosas.

#### 

#### Research and explore

Al implementar una función que requiere que use una tecnología con la que ambos no están familiarizados, primero tendrá que investigar y explorar. Este trabajo no encaja en los enfoques nítidos de "navegador-conductor" o "ping-pong". Por ejemplo, navegar por los resultados del motor de búsqueda juntos en la misma pantalla no suele ser muy eficaz.

#### 

#### Documentation

Otra cosa en la que trabajar juntos más allá del código es la documentación. Reflexionen juntos si hay alguna documentación necesaria para lo que han hecho. Nuevamente, dependiendo del caso en cuestión y sus preferencias individuales, puede crear la documentación juntos o hacer que uno de ustedes la cree, luego el otro la revise y la redacte.

La documentación es un gran ejemplo de una tarea en la que una pareja puede mantenerse disciplinada. A menudo es una tarea que se deja para el final, y cuando es lo último que nos impide tener la gran sensación de poner nuestra historia en "Terminado", la mayoría de las veces la saltamos o "improvisamos". Trabajar en pareja nos mantiene honestos acerca de algunas de las cosas valiosas, pero molestas, por las que estaremos muy agradecidos en el futuro.

## 

## BIBLIOGRAFÍA

On pair programming. (s. f.). martinfowler.com.

<https://martinfowler.com/articles/on-pair-programming.html>