

Programación en pareja

umh2818-TADS



Programación en pareja

La **Programación en Pareja** (o *Pair Programming* en inglés) requiere que dos programadores participen en un esfuerzo combinado de desarrollo en un sitio de trabajo. Cada miembro realiza una acción que el otro no está haciendo actualmente: Mientras que uno codifica las pruebas de unidades el otro piensa en la clase que satisfará la prueba, por ejemplo.

La persona que está haciendo la codificación se le da el nombre de *controlador* mientras que a la persona que está dirigiendo se le llama el *navegador*. Se sugiere a menudo para que a los dos socios cambien de papeles por lo menos cada media hora o después de que se haga una prueba de unidad.

1 Ventajas

La programación en pareja se enfoca en las siguientes ventajas, ordenadas de mayor a menor.

- Más Disciplina. Emparejando correctamente es más probable que hagan “lo que se debe hacer” en lugar de tomar largos descansos.
- Mejor código. Emparejando similares es menos probable producir malos diseños ya que su inmersión tiende a diseñar con mayor calidad.
- Flujo de trabajo constante. El emparejamiento produce un flujo de trabajo distinto al trabajar solo. En pareja el flujo de trabajo se recupera más rápidamente: un programador pregunta al otro “¿por dón-

de quedamos?”. Las parejas son más resistentes a las interrupciones ya que un desarrollador se ocupa de la interrupción mientras el otro continúa trabajando.

- Múltiples desarrolladores contribuyen al diseño. Si las parejas rotan con frecuencia en el proyecto significa que más personas están involucradas con una característica en particular. Esto ayuda a crear mejores soluciones, especialmente cuando una pareja no puede resolver un problema difícil.
- Moral mejorada. La programación en parejas es más agradable para algunos programadores, que programar solos.
- Propiedad colectiva del código. Cuando el proyecto se hace en parejas, y las parejas se rotan con frecuencia, todos tienen un conocimiento del código base.
- Enseñanza. Todos, hasta los novatos, poseen conocimientos que los otros no. La programación en pareja es una forma amena de compartir conocimientos.
- Cohesión de equipo. La gente se familiariza más rápidamente cuando programa en pareja. La programación en pareja puede animar el sentimiento de equipo.
- Pocas interrupciones. La gente es más renuente a interrumpir a una pareja que a una persona que trabaja sola.
- Menos estaciones de trabajo. Ya que dos personas van a trabajar en una estación de trabajo, se requieren menos estaciones de trabajo, y las estaciones extras pueden ser ocupadas para otros propósitos.

Los estudios han demostrado que después de entrenar para las “*habilidades sociales*” implicadas, parejas de programadores son más de dos veces más productivos que uno para una tarea dada. Según *The Economist*:

“Laurie Williams de la universidad de Utah en Salt Lake City ha demostrado que los programadores emparejados son solamente 15% más lentos de dos programadores trabajando independientemente, pero producen 15% menos errores. Y ya que la prueba y depuración son a

menudo muchas veces más costosa que la programación inicial, esto es da un resultado impresionante”

(Dr. Williams is currently Assistant Professor of Computer Science at **North Carolina State University**)

Un estudio reciente sobre un experimento rigurosamente científico en el cual se compara parejas de novatos contra novatos solos son mucho más productivos que parejas de expertos contra expertos solos, de acuerdo con “Int J. of Human Computer Studies Vol (64) 2006”

2 Críticas

- Desarrolladores expertos pueden encontrar tedioso enseñar a un desarrollador menos experimentado en un ambiente emparejado.
- Muchos desarrolladores prefieren trabajar solos y encuentran el ambiente emparejado incómodo.
- La productividad es difícil de medir, para comparar desarrolladores en parejas contra desarrolladores trabajando solos, usan métricas de programación que son controvertidas en el mejor de los casos.
- Diferencias en el estilo de codificación pueden resultar en conflictos, aunque esto a su vez promueve la normalización (estandarización), para que todo el mundo pueda entender el código.
- En caso de que la pareja tenga cronogramas de trabajo ligeramente distintos, lo cual es común en un ambiente que valora el balance trabajo - vida, la pareja solo funcionara en la intersección de los cronogramas. Por lo tanto, se requieren más horas persona para completar una tarea, un día típico tiene menos horas en pareja, lo cual incrementa el tiempo para completar la tarea.
- Una compañía que funciona con Teletrabajo (trabajo desde el hogar o a distancia) o cuando un empleado debe trabajar desde afuera de las oficinas por cualquier razón, la programación en pareja se hace difícil o hasta imposible.

3 Véase también

- **Programación Extrema**

4 Enlaces externos

- VS Anywhere <https://vsanywhere.com/web/>

5 Origen del texto y las imágenes, colaboradores y licencias

5.1 Texto

- **Programación en pareja** *Fuente:* https://es.wikipedia.org/wiki/Programaci%C3%B3n_en_pareja?oldid=87460001 *Colaboradores:* Boticario, Viko~eswiki, RobotQuistnix, Yrbot, CEM-bot, Especiales, TXiKiBoT, SieBot, LordKynos, Nicop, Farisori, Alexbot, Diegusjaimes, Arjuno3, Xqbot, EmausBot, TuHan-Bot, ChuispastonBot, WikitanvirBot, !Silent, KLBot2, Fernandoquadra, Vsanywhere y Anónimos: 10

5.2 Imágenes

- **Archivo:Pair_Programming.jpg** *Fuente:* https://upload.wikimedia.org/wikipedia/commons/6/66/Pair_Programming.jpg *Licencia:* CC BY-SA 3.0 *Colaboradores:* Trabajo propio *Artista original:* Calqui

5.3 Licencia del contenido

- Creative Commons Attribution-Share Alike 3.0