

## Parte II – Prácticas genéricas del desarrollo Ágil

### Tema 3: Programación en parejas

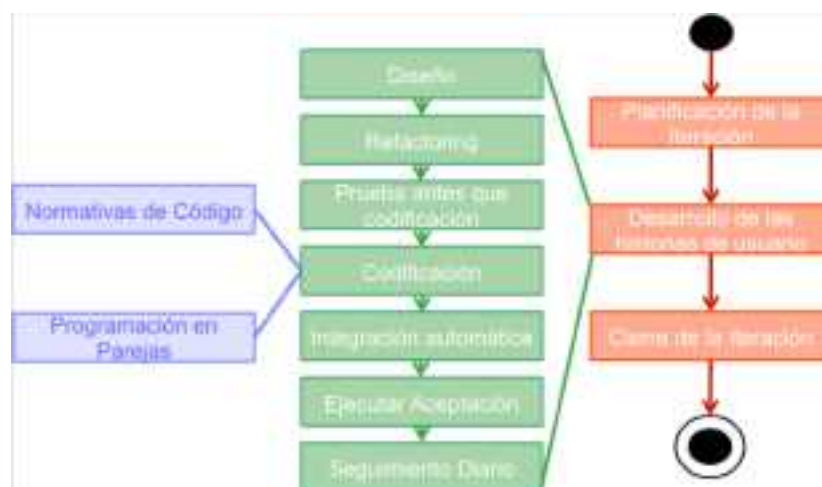
Programación en parejas

Normativas de código → Usaremos "checkStyle"  
 - Generar ~ 20 y 21 feb  
 - Aplicarlo ~ 17 y 18 feb  
 Propiedad Colectiva Código → Control de versiones (Gitlab - SourceTree)  
 Integración Automática → Integración Automática de Maven ~ 17 y 18 feb  
 ~ 20 y 21 feb



- Positivo
- Respeto
  - Niveles de conocimiento similares.
  - Buena sinergia (buscando y escribiendo)

## Funcionamiento interno de una iteración



- Negativo
- No han definido sus tareas.
  - Se han puesto a programar directamente

Problemas programación en parejas:

Uno tiene que seguir su diseño planeado, no cambiarlo por el del observador.

Día de trabajo uni:

9:00 - 13:00 Clase

13:00 - 14:00 Comida

14:00 - 15:00 Trabajo en pareja.  
Incidente aula

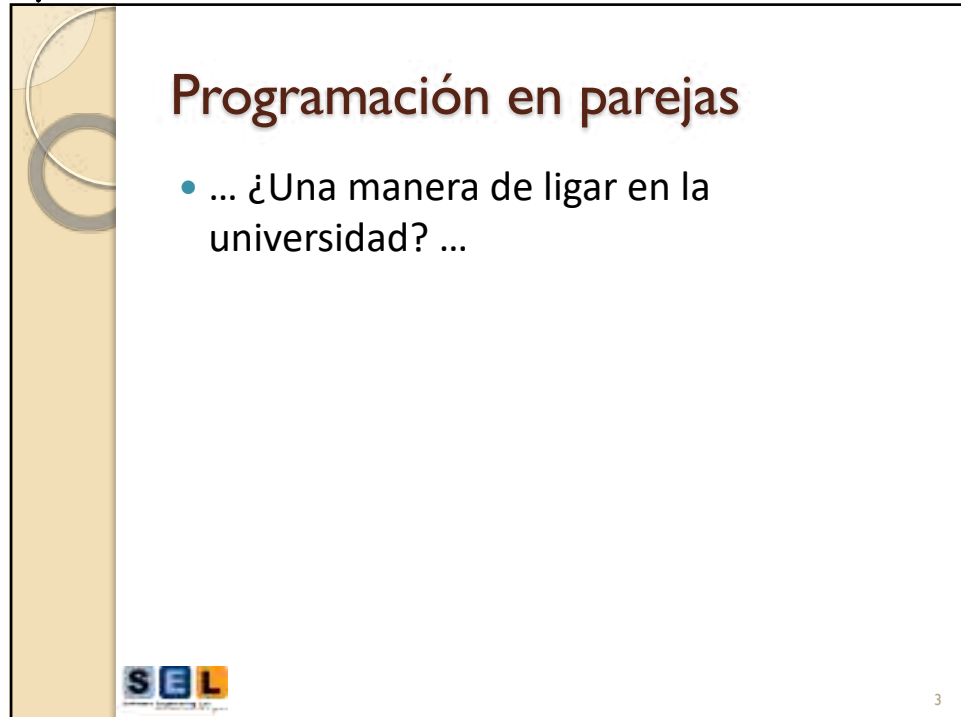
15:00 - 17:00 A casa

17:00 - 19:00 Trabajo en casa (individual)

19:00 - 20:00 whatsapp, Skype ~ paer en comun.

No es bided para Trab. en parejas.

11/2/19

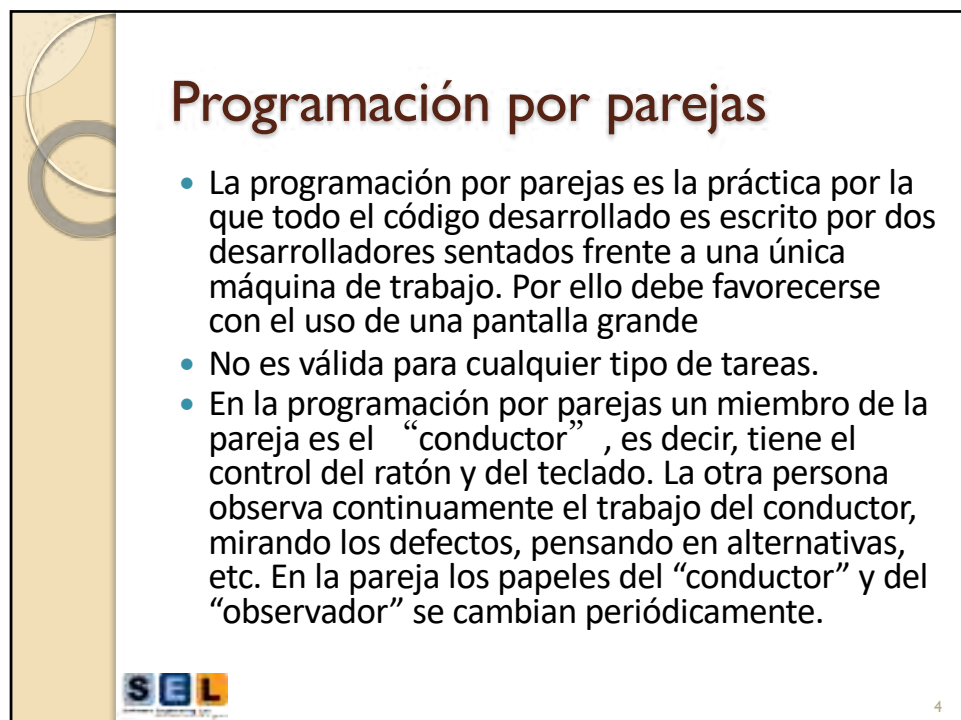


## Programación en parejas

- ... ¿Una manera de ligar en la universidad? ...

SEL

3



## Programación por parejas

- La programación por parejas es la práctica por la que todo el código desarrollado es escrito por dos desarrolladores sentados frente a una única máquina de trabajo. Por ello debe favorecerse con el uso de una pantalla grande
- No es válida para cualquier tipo de tareas.
- En la programación por parejas un miembro de la pareja es el "conductor", es decir, tiene el control del ratón y del teclado. La otra persona observa continuamente el trabajo del conductor, mirando los defectos, pensando en alternativas, etc. En la pareja los papeles del "conductor" y del "observador" se cambian periódicamente.

SEL

4

1) Preparación: Definir como vamos a hacer

Codificar

Interfaz

↕

Revisar como lo vamos a probar.

Dividir el trabajo:

API

REST

→ Objeto con interfaces y de servidor funciones

Funcionalidades → Escribir el código...

Pruebas → Proporciona apoyos restricciones y planis pruebas.

2) Trabajo individual:

===

3) Cierre del trabajo en pareja.

Poner el código funcional y ejecutar las pruebas, para comprobar las funcionalidades.  
Ver si sigue la normativa y explorar como va la noche.

11/2/19

## Programación en parejas

- El emparejamiento es dinámico, dos personas pueden estar emparejadas por la mañana y por la tarde tener otras parejas.
- Los integrantes deben tener niveles de experiencia similar.
- En principio escribir todo el código por parejas sería más lento, a no ser que:
  - Los estándares de codificación reduzcan las disputas.
  - Cada uno este fresco y descansado y así evitar las discusiones absurdas.
  - Las parejas escriben las pruebas juntas, dando la posibilidad de alinear su comprensión antes de afrontar el meollo de la implementación.
  - Las parejas tengan la metáfora para fundamentar sus discusiones sobre los nombres y el diseño básico.
- Las parejas estén trabajando sobre diseños sencillos.



5

## Ventajas

- Se genera código con menos defectos no encontrados.
  - Por tanto se consigue más calidad
- Se finaliza más rápido, porque se detectan y corrigen los defectos más rápidamente
- Favorece la comunicación entre los participantes y con otros miembros.
- Favorece la revisión conjunta de código y pruebas.
- El código es conocido por más de un ingeniero.
- La utilización de un estándar de código facilita el trabajo
- Se crea un sentimiento de compromiso y propiedad colectiva de código.



Principios de Desarrollo de Software – Parte I - Aspectos Éticos y Legales en la Profesión de Ingeniero de Software

6

## Lecciones aprendidas (I)

- En la programación por parejas, dos programadores producen un artefacto (diseño, algoritmo, código, etc).
  - Ambos son un organismo coherente, inteligente que trabaja como una mente, responsable de cada aspecto que pueda surgir.
- Idealmente, los programadores deben trabajar juntos constantemente.
  - Sin embargo, la realidad es que la pareja por diferentes causas (enfermedad, tiempo, etc.) puede llegar a trabajar de forma individual.
  - En algunos equipos ágiles es posible en una primera fase determinar qué trabajo se puede hacer por separado, para luego revisarlo juntos.



7

## Lecciones aprendidas (II)

- Mientras un miembro de la pareja se encarga de escribir el otro piensa estrategias de las implicaciones del diseño y puede realizar una revisión del diseño continua.
- Cuando un grupo entero adopta la programación por parejas como la manera normal de trabajo, la continuidad a largo plazo de cualquier pareja particular llega a ser menos importante.
  - Intercambiando la pareja regularmente en un grupo, un programador individual puede mantener suficiente conocimiento general para sustituir, si un miembro falta en un determinado momento.



8

## Lecciones aprendidas (III)

- Existen tres razones por las que la programación por parejas es más valorada en la industria:
  - Los progresos con más errores pasan más tiempo corrigiéndose.
  - Hay un coste de calidad a largo plazo asociado a los diseños confusos que son más difíciles de cambiar.
  - La calidad del código es más alta aunque no sea productivo.
- Con la programación por parejas se mejora la calidad del software y la productividad. Mejora la satisfacción profesional y confianza total.



9

## Programación en Parejas

Un Caso Práctico



## Introducción del Caso Práctico

- La mayor parte de la gente que no ha probado la programación por parejas la rechazan.
- A través del caso práctico que se describe se va a demostrar problemas y resultados comunes de la programación en parejas.



11

## Experimento Universitario (I)

- En 1999 en la Universidad de Utah, los estudiantes del curso Senior Software Engineering participaron en un experimento estructurado.
- Los estudiantes fueron informados de la importancia de guardar la información necesaria del experimento.
- Todos los estudiantes atendieron a las mismas clases, recibieron la misma instrucción y participaron en discusiones sobre los pros y los contra de la programación por parejas.
- Cuando se les preguntó el primer día de clase el 85% de los estudiantes (35 de los 41) indicó su preferencia por la programación por parejas. Más adelante muchos de aquellos estudiantes admitieron que inicialmente eran renuentes, pero curiosos.



12

## Experimento Universitario (II)

- Se formaron dos grupos.
  - Trece estudiantes formaron el grupo de control en el que todos trabajaban individualmente en todas los programas.
  - Veintiocho estudiantes formaron el grupo experimental en el que todos trabajaban en equipos de colaboración de dos personas.
    - Las parejas de colaboración también hicieron asignaciones adicionales para distribuirse la carga del trabajo entre los dos.



13

## Experimento Universitario (III)

- Antes de realizar el curso los estudiantes tenían práctica significativa en la codificación.
- La duración del ciclo, la productividad y los resultados de calidad fueron comparados entre los dos grupos.
  - Los estudiantes registraron la información sobre el tiempo que pasaron en el proyecto en una herramienta web.



14

## Resultados (I)

- Existe un periodo inicial de ajuste de transición, que puede durar horas o días.
- Los programadores experimentados en trabajar por parejas dan la prioridad a:
  - Qué partes del ciclo de desarrollo son las más importantes para trabajar conjuntamente,
  - Qué se puede hacer por separado,
  - Qué hacer con el trabajo independientemente desarrollado al juntarse.



15

## Resultados (II)

- Los programadores ven el análisis y el diseño por parejas más crítico que la implementación por parejas.
- La mayoría de los programadores realizan una revisión cuidadosa del trabajo individual y lo incorporan en el proyecto.
- Una minoría pequeña integra el trabajo individual sin revisar.



16



## Resultados (III)

- El caso de estudio produjo unos resultados cuantitativos que apoyan los de la programación por pareja en industria.
- Los estudiantes realizaron cuatro programas durante seis semanas.
- Trece individuales y catorce parejas colaborando completaron cada programa.
- Los que trabajaron en parejas pasaron más casos de pruebas automatizadas que los que trabajaron individualmente. Esto se puede ver en la siguiente tabla.

	Individualmente	En parejas
Programa 1	73,4%	86,4%
Programa 2	78,1%	88,6%
Programa 3	70,4%	87,1%
Programa 4	78,1%	94,4%



17

## Resultados (IV)

- En un principio se rechaza la programación por parejas porque se asume que habrá un 100% de horas por programador de aumento, poniendo dos programadores en un trabajo que uno puede hacer. Después del periodo inicial de ajuste, las horas por programador se redujeron.



18

## Resultados (V)

- La colaboración mejora el proceso, ya que “dos cabezas es mejor que una” y “cuatro ojos son mejores que dos” .
- Las pruebas en parejas es la parte menos crítica del ciclo de desarrollo, siempre y cuando la pareja desarrolle el caso de prueba junta.
- Algunas parejas se dividen los casos de prueba y cuando encuentran los defectos, colaboran para encontrar una mejor solución.



19

## Resultados (VI)

- Los 41 programadores implicados en el experimento de la universidad fueron encuestados seis veces.
- El 96% indicó que fue más productivo el trabajo en parejas, que cuando se trabaja de forma individual.
- El 95% de los estudiantes indicó que las soluciones desarrolladas en parejas eran más fiables que las individuales.



20