**Instituto Tecnológico de Estudios Superiores de Monterrey**

****

**Reporte Final**

**Calidad y pruebas de software**

Por:

Gerardo Aldair Ponce Gomez - A00818934

Contenido

[Introducción 3](#_Toc7899289)

[Análisis de la Estimación de Tamaño 3](#_Toc7899290)

[1. ¿Cuál fue el tamaño (LDC AyM) de cada uno de tus programas? ¿Cuál fue tu promedio de LDC AyM desarrolladas por programa? 3](#_Toc7899291)

[2. ¿Cuál fue el programa con más LDC AyM y cuál el que tuvo menos? ¿Por qué? 3](#_Toc7899292)

[3. ¿Cuál fue el % de error en la estimación de tamaño (real/plan-1) en tus pr ogramas 2 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tamaño de estos 6 programas juntos? 4](#_Toc7899293)

[4. ¿Cuál fue el programa con mayor error de estimación de tamaño y cuál el menor? ¿Por qué? 4](#_Toc7899294)

[5. En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿el tamaño real de LDC AyM estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó? 5](#_Toc7899295)

[6. Basado en los datos históricos que has analizado sobre el error de estimación del tamaño, ¿Qué meta realista (alcanzable) te puedes poner para mejorar? 5](#_Toc7899296)

[7. ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta? 5](#_Toc7899297)

[Análisis de la Estimación de Tiempo 5](#_Toc7899298)

[1. ¿Cuál fue el tiempo en horas dedicado a cada uno de tus programas? ¿Cuál fue tu promedio de horas dedicadas por programa? 5](#_Toc7899299)

[2. ¿Cuál fue el programa al que le dedicaste más tiempo y a cuál menos? ¿Por qué? 6](#_Toc7899300)

[3. ¿Cuál fue el % de error en la estimación de tiempo (real/plan-1) en tus programas 1 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tiempo de estos 7 programas juntos? 6](#_Toc7899301)

[4. ¿Cuál fue el programa con mayor error de estimación de tiempo y cuál el menor? ¿Por qué? 7](#_Toc7899302)

[5. En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿El tiempo real estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó? 8](#_Toc7899303)

[6. ¿Cuál fue la productividad de cada uno de tus programas? ¿Cuál fue tu productividad promedio? 8](#_Toc7899304)

[7. ¿Cuál es la tendencia de tu productividad? ¿Va a la baja, o al alta? ¿Es inestable? ¿Qué puedes hacer para mejorarla? 8](#_Toc7899305)

[8. ¿Qué tanto fueron afectados los errores de estimación de tiempo por tus errores de estimación del tamaño? 9](#_Toc7899306)

[9. Basado en los datos históricos que has analizado sobre el error de estimación del tiempo, ¿Qué meta realista (alcanzable) te puedes poner para mejorar? 9](#_Toc7899307)

[10. ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta? 9](#_Toc7899308)

[Análisis de la Calidad (Defectos) 9](#_Toc7899309)

[1. ¿Qué tipo de defecto es al que más tiempo le dedicaste en la fase de compilación? ¿Y en la fase de pruebas? 9](#_Toc7899310)

[2. ¿Qué tipo de defecto es el que más inyectaste en la fase de Diseño? ¿Y en la fase de Codificación? 10](#_Toc7899311)

[3. ¿Cuál es el promedio de horas por defecto dedicadas al remover defectos en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas? 10](#_Toc7899312)

[4. ¿Cuál es la tendencia en los 7 programas del total de defectos inyectados por KLDC? 10](#_Toc7899313)

[5. ¿Cuál es la tendencia en los 7 programas de los defectos removidos por KLDC en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas? 11](#_Toc7899314)

[6. ¿Cómo se comparan las tasas de remoción de defectos (defectos removidos por hora) en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas en los 7 programas? 13](#_Toc7899315)

[7. ¿Cuál fue tu velocidad (LDC por hora) a la que revisaste el código de cada programa? 13](#_Toc7899316)

[8. ¿Cuál fue la palanca de remoción de defectos (DRL) de cada programa para las fases de Revisión del Diseño y Revisión del Código? ¿Es bueno o malo? 13](#_Toc7899317)

[9. ¿Existe alguna relación entre el Yield de cada programa y la velocidad (LDC por hora) a la que revisaste el código? 14](#_Toc7899318)

[10. ¿Existe alguna relación entre el Yield de cada programa y su A/FR? 14](#_Toc7899319)

[11. ¿Estás encontrando los defectos en las fases de Revisión del Diseño y del Código, o se te están escapando a las fases de Compilación y Pruebas? ¿Por qué? 15](#_Toc7899320)

[12. Basado en los datos históricos que has analizado sobre la calidad de tus programas, ¿Qué meta realista (alcanzable) te puedes poner para mejorar? 15](#_Toc7899321)

[13. ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta? 15](#_Toc7899322)

[Conclusión: 15](#_Toc7899323)

[Párrafo de honestidad: 15](#_Toc7899324)

# Introducción

Este reporte tiene como objetivo el mostrar los datos mostrados y el hacer un análisis de los resultados obtenidos durante todo el semestre, Comparando diferentes datos y mostrando otros con el objetivo de encontrar el área de oportunidad de cada una. Se tomaron en cuenta diferente valores tales como las LDC que se desarrollaron en el transcurso del semestre.

De igual manera se explicara diferentes resultados en las que se expondrán graficas para corroborar los datos expuestos junto a breves explicaciones sobre cada una de ellas, donde se analizaran tendencias mostradas durante el semestre

# Análisis de la Estimación de Tamaño

## ¿Cuál fue el tamaño (LDC AyM) de cada uno de tus programas? ¿Cuál fue tu promedio de LDC AyM desarrolladas por programa?

* Programa 1
  + LDC: 109
  + AyM: 109
  + Promedio: 109
* Programa 2
  + LDC: 264
  + AyM: 157
  + Promedio: 210.5
* Programa 3
  + LDC: 172
  + AyM: 111
  + Promedio: 141.5
* Programa 4
  + LDC: 116
  + AyM: 86
  + Promedio: 101
* Programa 5
  + LDC: 114
  + AyM: 24
  + Promedio: 69
* Programa 6
  + LDC: 379
  + AyM: 138
  + Promedio: 258.5
* Programa 7
  + LDC: 290
  + AyM: 222
  + Promedio: 256

## ¿Cuál fue el programa con más LDC AyM y cuál el que tuvo menos? ¿Por qué?

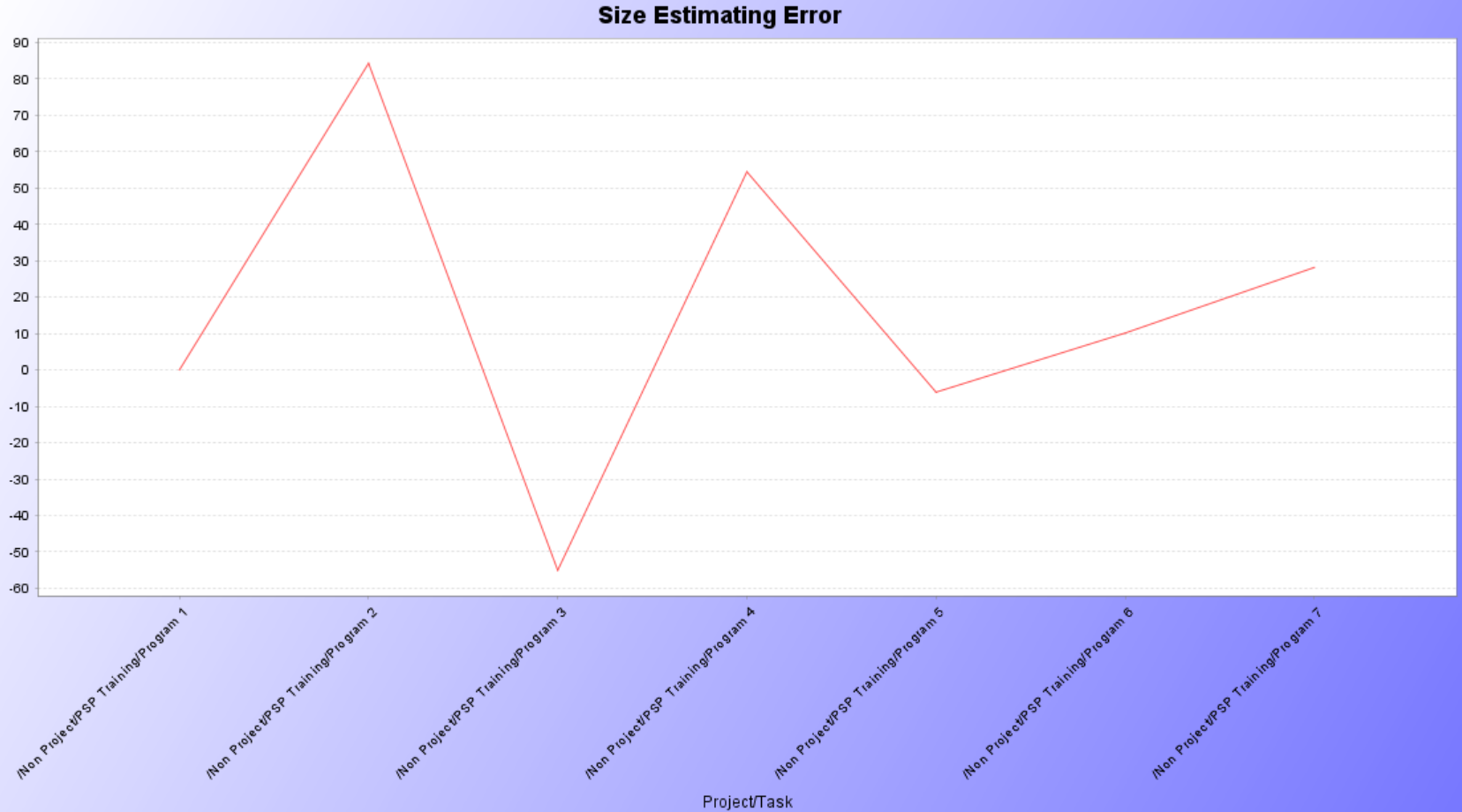
* Programa con más LDC:
  + Programa 6 con 379
* Programa con más AyM:
  + Programa 7 con 222
* Programa con menos LDC:
  + Programa 1 con 109
* Programa con menos AyM:
  + Programa 5 con 24
* Programa con mas promedio entre LDC y AyM
  + Programa 6 con 258.5 de promedio
* Programa con menos promedio entre LDC y AyM
  + Programa 5 con 69 de promedio

El programa con mas LDC fue el 6 y se debió a que fue un programa en el que se tuvieron que reutilizar muchas líneas de codigo de laboratorios pasados, ya que se hacían cálculos previamente hechos y solo se tuvieron que agregar pocas líneas de codigo debido a que los cálculos faltantes no eran muchos. En cambio, el programa con mas AyM fue el 7 debido a que fue un programa que casi no utilizaba codigo previamente hecho, por lo que se tuvo que hacer un codigo nuevo que cumpliera con los requerimientos del programa.

En cambio, el que obtuvo menos LDC fue el 1 ya que fue el programa mas pequeño de todos los que desarrollamos, y el de menos AyM fue el 6 ya que mucho codigo fue hecho desde 0 y no se utilizo casi nada de codigo rehusado.

El programa con mas LDC y AyM fue el 6 debido a que rehusó mucho codigo, como tambien se desarrollo para el programa, en cambio el de menos fue el 5 ya que en este caso se creo mas codigo del que se rehusó

## ¿Cuál fue el % de error en la estimación de tamaño (real/plan-1) en tus pr ogramas 2 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tamaño de estos 6 programas juntos?



* ***Programa 2***
  + Error: 84.3%
* ***Programa 3***
  + Error: -55%
* ***Programa 4***
  + Error: 54.5%
* ***Programa 5***
  + Error: -6.1%
* ***Programa 6***
  + Error: 10.2%
* ***Programa 7***
  + Error: 28.2%
* ***Total***
  + Error: 14.68%

## ¿Cuál fue el programa con mayor error de estimación de tamaño y cuál el menor? ¿Por qué?

El de mayor error fue el programa 2 ya que ahí todavia no sabia cuando tardaba o que tan bien o mal desarrollaba asi que lo hice imaginándome mas o menos cuanto tiempo me costaría hacerlo según como me percibía, en cambio el que menos porcentaje de error tuvo fue el programa 5, ya que aquí ya tenia experiencia y sabia de la calidad de mi programación, y lo que ayudo en este fue que este era el programa mas pequeños y por lo que fue mas facil de calcular ya que era mas facil visualizar las cosas que tenia que tener el programa

## En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿el tamaño real de LDC AyM estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó?

En este caso no tengo la información requerida ya que nunca utilice los métodos A o B del probe wizard, por lo que no dispongo información

## Basado en los datos históricos que has analizado sobre el error de estimación del tamaño, ¿Qué meta realista (alcanzable) te puedes poner para mejorar?

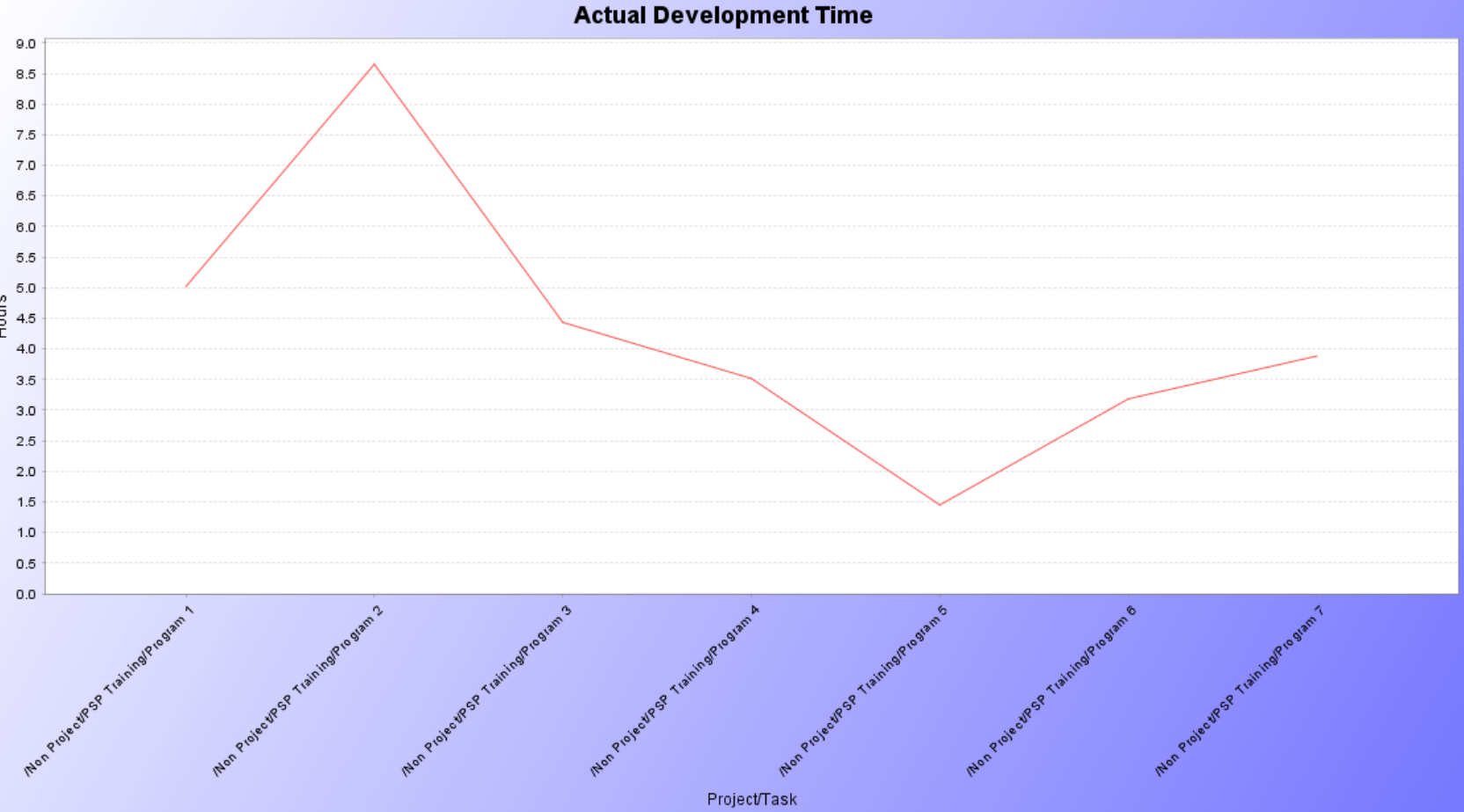
Mi meta seria el de poder tener una estimación en la cual tenga un margen de errr +-15% , esto seria en corto y mediano plazo, ya que a futuro aspiro a poder hacer predicciones mas exactas sobre mis trabajos logrando asi tener mejor planeación en cada uno de ellos

## ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta?

Empezar a documentarme de las diferentes herramientas o técnicas que puedo utilizar para poder desarrollar el programa, ya que eso haria que pueda saber con exactitud cuantas líneas podria realizar, y no solo hacer un estimado pensando de que con tal técnica creo que hare n líneas de codigo y otra muy importante, es el de empezar a realizar esta actividad en cada codigo mio, ya que asi podria estar practicando constantemente y poder tener mas experiencia y conocimiento sobre mi mismo.

# Análisis de la Estimación de Tiempo

## ¿Cuál fue el tiempo en horas dedicado a cada uno de tus programas? ¿Cuál fue tu promedio de horas dedicadas por programa?



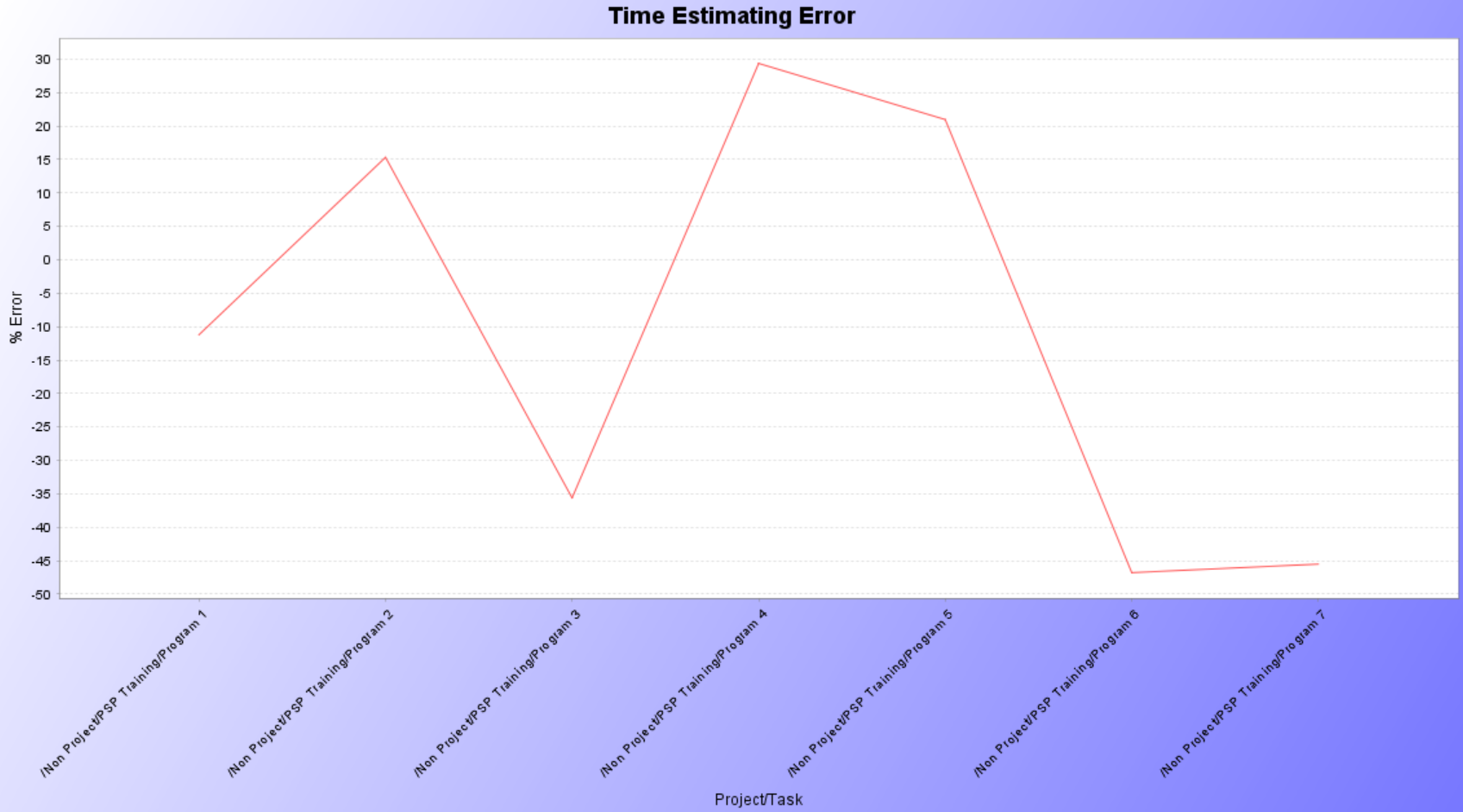
* ***Programa 1***
  + Tiempo: 5.02
* ***Programa 2***
  + Tiempo: 8.65
* ***Programa 3***
  + Tiempo: 4.43
* ***Programa 4***
  + Tiempo: 3.52
* ***Programa 5***
  + Tiempo: 1.45
* ***Programa 6***
  + Tiempo: 3.18
* ***Programa 7***
  + Tiempo: 3.88
* ***Total***
  + Tiempo total: 30.08
  + Promedio: 4.30

## ¿Cuál fue el programa al que le dedicaste más tiempo y a cuál menos? ¿Por qué?

Al programa que le dedique mas tiempo fue al programa 2 y esto se debió principalmente a que malinterprete los requerimientos y no comprendí algunos de ellos, ya que al no poner a analizarlo al fondo y empezar a codificar lo mas antes posibles no tuve completo entendimiento de que era lo que me pedían, y muchas veces lo que hacia era que me daban cálculos erróneos por no saber como se calcula bien las líneas de codigo.

Al programa que menos le dedique fue al 5 ya que este era el mas pequeño de todos y por que tambien ya empezaba a leer cada vez mas los requerimientos para evitar equivocarme y en este caso solo tuve que realizar el codigo una vez junto con los cálculos, por lo que fue mas rapido de lo que normalmente seria.

## ¿Cuál fue el % de error en la estimación de tiempo (real/plan-1) en tus programas 1 al 7? En promedio, ¿Cuál fue el % de error en la estimación de tiempo de estos 7 programas juntos?



* ***Programa 1***
  + Error: -11.2%
* ***Programa 2***
  + Error: 15.3%
* ***Programa 3***
  + Error: -35.6%
* ***Programa 4***
  + Error: 29.4%
* ***Programa 5***
  + Error: 21%
* ***Programa 6***
  + Error: -46.8%
* ***Programa 7***
  + Error: -45.5%
* ***Promedio***
  + Plan: 37.04 h
  + Actual: 30.08
  + Error: 0.83%

## ¿Cuál fue el programa con mayor error de estimación de tiempo y cuál el menor? ¿Por qué?

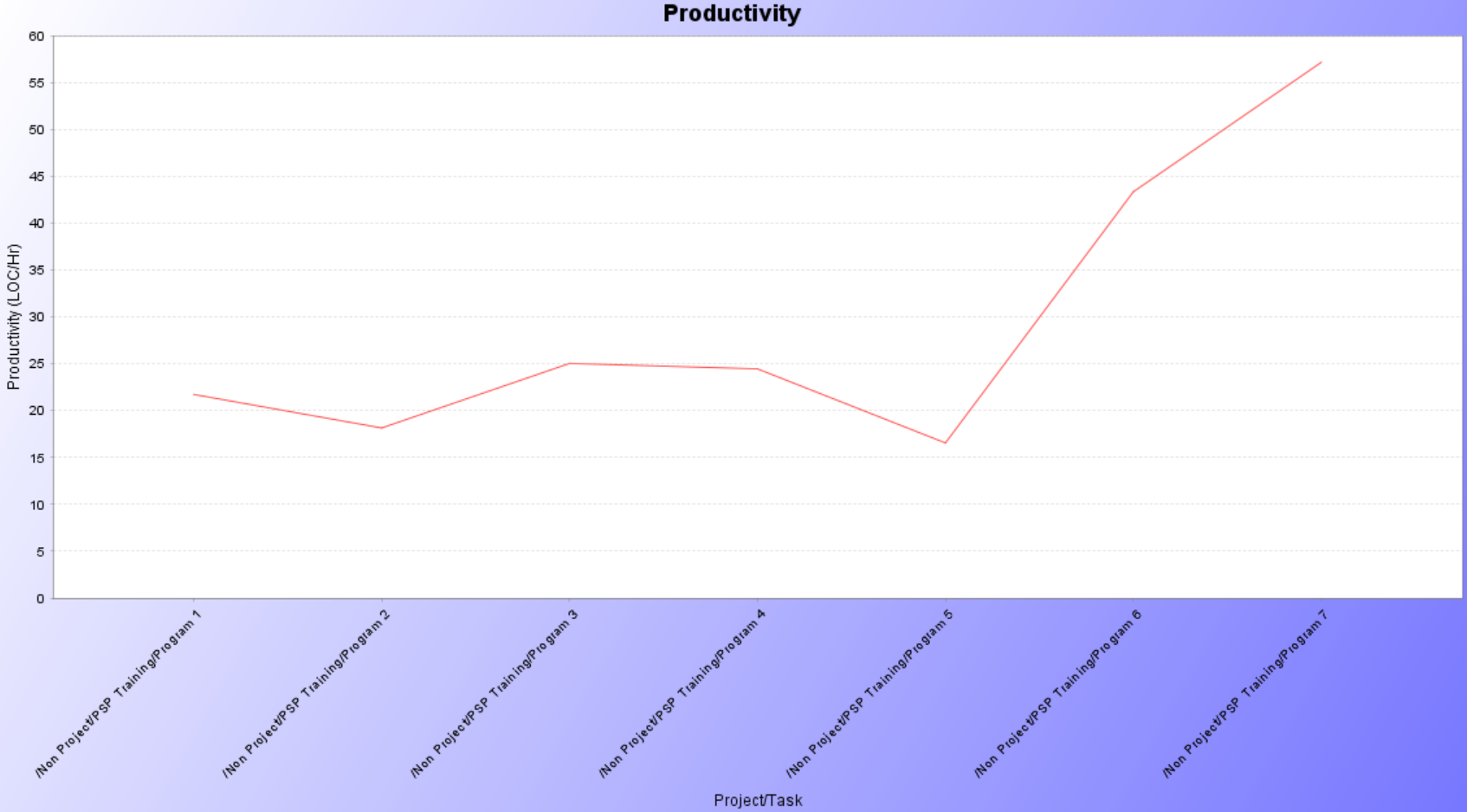
El programa con mayor error fue el programa 3, ya que en este caso ya que aquí, aunque las líneas de codigo no fueron tantas malinterprete la forma de hacer los cálculos, por lo que estuve mucho tiempo intentado averiguar como era que se tenia que hacer el codigo para poder obtener los cálculos necesarios, por eso aunque estimaba que iba a tardar poco y al final no realice muchas líneas de codigo tarde en empezar a desarrollarlas de manera correcta

El programa con menor error fue el 1 debido a que cuando leí los requerimientos entendi lo que tenia que hacer y se parecía a programas que ya habia hecho anteriormente por lo que al tener ya algo de experiencia fue donde mejor me salio la estimación.

## En aquellos programas donde pudiste utilizar el método PROBE “A” o el “B”, ¿El tiempo real estuvo dentro del intervalo de predicción estadístico del 70%? Si no estuvo, ¿Qué pasó?

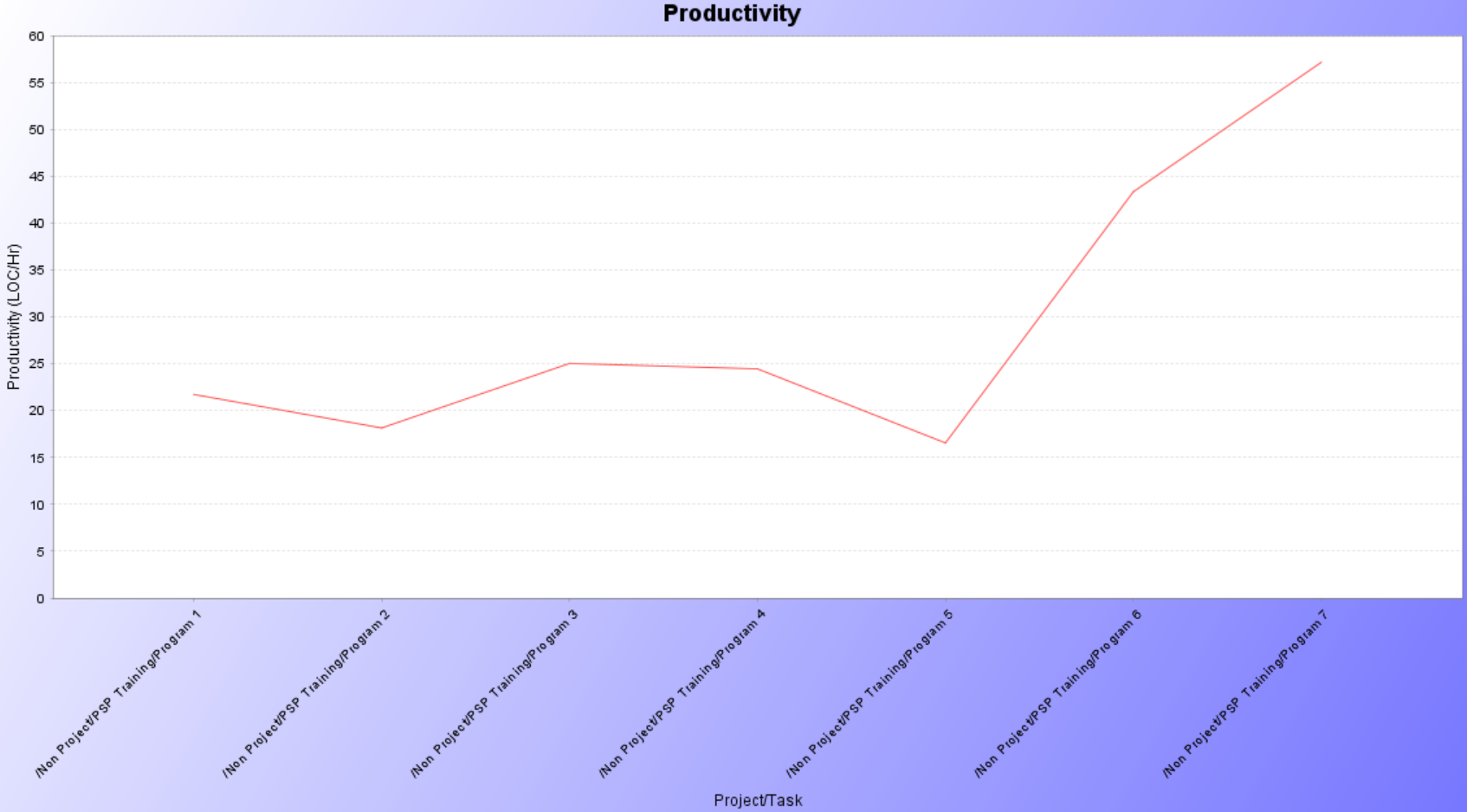
En este caso no tengo la información requerida ya que nunca utilice los métodos A o B del probe wizard, por lo que no dispongo información

## ¿Cuál fue la productividad de cada uno de tus programas? ¿Cuál fue tu productividad promedio?



* ***Programa 1***
  + Productividad: 21.7 (LOC/Hr)
* ***Programa 2***
  + Productividad: 18.2 (LOC/Hr)
* ***Programa 3***
  + Productividad: 25 (LOC/Hr)
* ***Programa 4***
  + Productividad: 24.5 (LOC/Hr)
* ***Programa 5***
  + Productividad: 16.6 (LOC/Hr)
* ***Programa 6***
  + Productividad: 43.4 (LOC/Hr)
* ***Programa 7***
  + Productividad: 57.2 (LOC/Hr)
* ***Promedio***
  + Productividad: 29.51 (LOC/Hr)

## ¿Cuál es la tendencia de tu productividad? ¿Va a la baja, o al alta? ¿Es inestable? ¿Qué puedes hacer para mejorarla?



La productividad va a la alta, por un tiempo estuvo estable en la que bajaba y subía muy poco, y fue al final donde a partir del programa 5, que fue la que obtuvo la productividad mas baja entre todos los laboratorios, despego y empezó a ir a la alta hasta alcanzar su pico en el programa 7, por lo que debo de evitar que baje demasiado y buscar que se mantenga estable entre los 25 y 30 LOC/Hr.

## ¿Qué tanto fueron afectados los errores de estimación de tiempo por tus errores de estimación del tamaño?

Fueron afectados, pero no tanto, ya que hubo unos casos donde mi porcentaje de error en el tamaño fue mucho mas grande de que mi porcentaje de error en el tiempo, y esto fue debido a lo que tardaba en comprender los requerimientos del programa, por lo que si hubo ocasiones en las que si me afecto pero en otras donde no

## Basado en los datos históricos que has analizado sobre el error de estimación del tiempo, ¿Qué meta realista (alcanzable) te puedes poner para mejorar?

Que puedo ser mas preciso con el tiempo que doy para acabar, m i meta seria un +-20% de error ya que es algo muy importante y para poder alcanzarla lo que necesito es seguir practicando con programas de diferentes tipos y agarrar mas experiencia, ya que la experiencia es una parte muy fundamental para poder dar correctamente bien un aproximado

## ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta?

Medir en todo momento los tiempos que me toma cada programa y cada fase, ya que asi sabre de verdad en que me equivoco y en cual estimo de mas o menos, por lo que a partir de ahora antes de empezar a codificar me voy a proponer un tiempo realista para acabar el codigo y anotar la diferencia en lo que tardo sobre lo que digo, y asi sucesivamente hasta ir perfeccionando la técnica

# Análisis de la Calidad (Defectos)

## ¿Qué tipo de defecto es al que más tiempo le dedicaste en la fase de compilación? ¿Y en la fase de pruebas?

Solo tuve un defecto en la fase de compilación y fue de tipo “Function”, como tambien en la fase de pruebas solo obtuve dos defectos y los dos tambien fueron del mismo tipo el cual fue “Function”.

## ¿Qué tipo de defecto es el que más inyectaste en la fase de Diseño? ¿Y en la fase de Codificación?

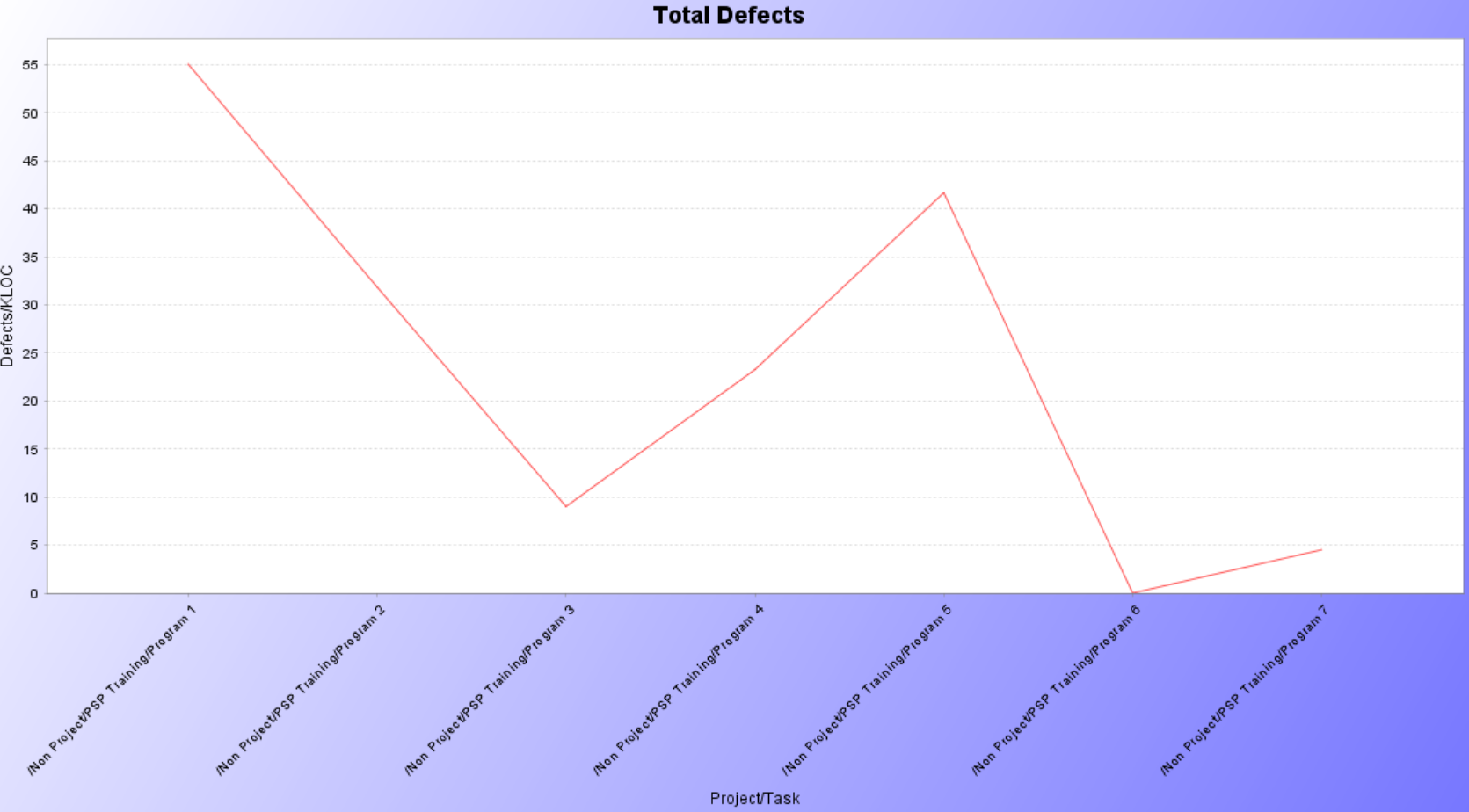
El tupo de defecto que mas inyecte en diseño fueron los de tipo “Function”, que estuvieron presente en 6 de los 9 errores, y estos eran generalmente errores por descuidos míos como el calcular mal o despejar de manera incorrecta en casos de cálculos matemáticos

El tipo de defecto que mas inyecto en la fase de codificación fueron los de tipo “Function” en los cuales se estuvo presente 4 de 7 veces ese tipo de defecto, que consistió varias veces llamadas mal a funciones o que hacían que el codigo no pudiera funcionar de manera correcta.

## ¿Cuál es el promedio de horas por defecto dedicadas al remover defectos en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas?

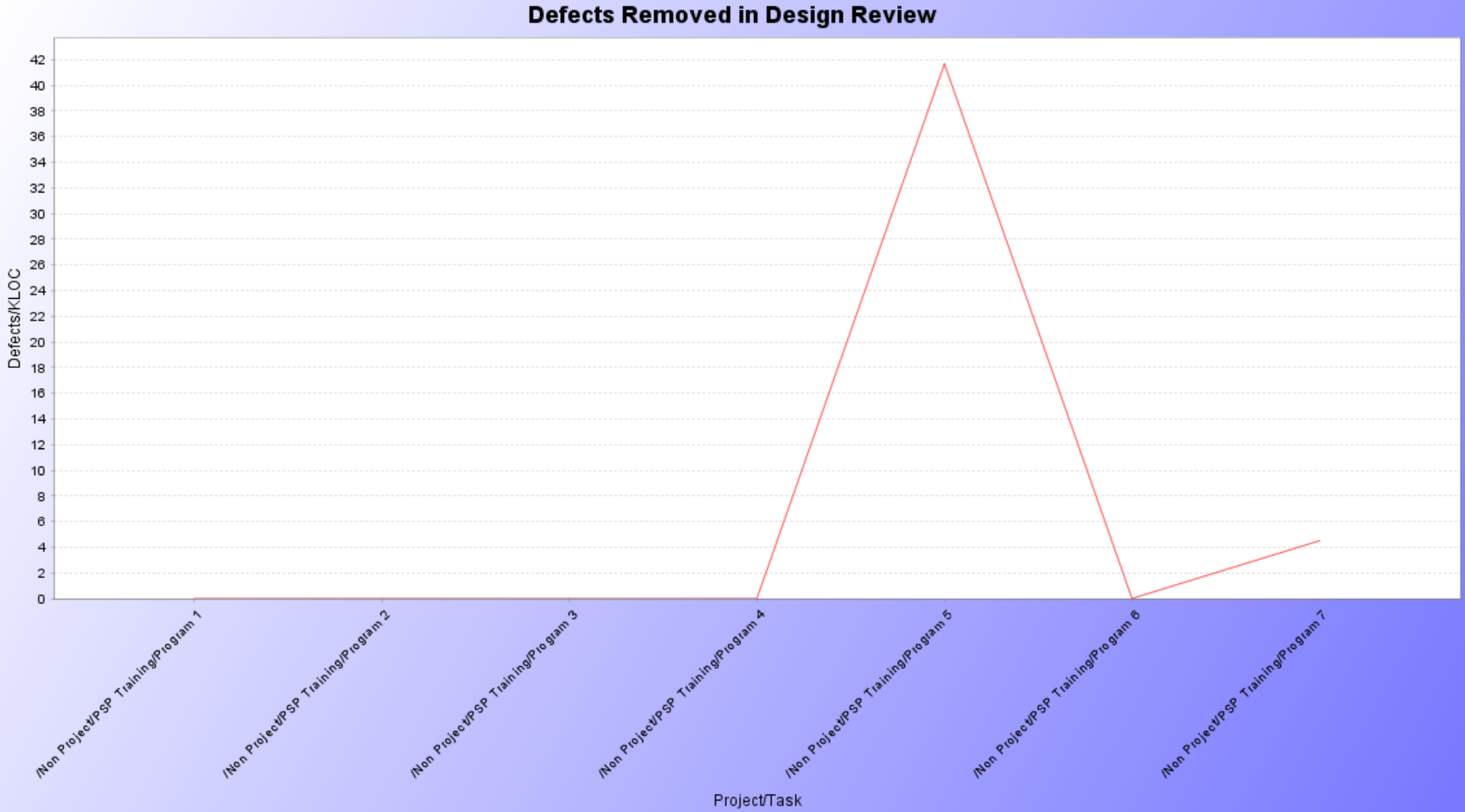
* ***Revisión de diseño***
  + Tiempo: 4.8
  + Defectos: 2
  + Promedio: 3.4
* ***Revisión de codigo***
  + Tiempo: 1.9
  + Defectos: 1
  + Promedio: 1.9
* ***Compilación***
  + Tiempo: 21.2
  + Defectos: 1
  + Promedio: 21.2
* ***Pruebas***
  + Tiempo: 39.2
  + Defectos: 2
  + Promedio: 19.6

## ¿Cuál es la tendencia en los 7 programas del total de defectos inyectados por KLDC?

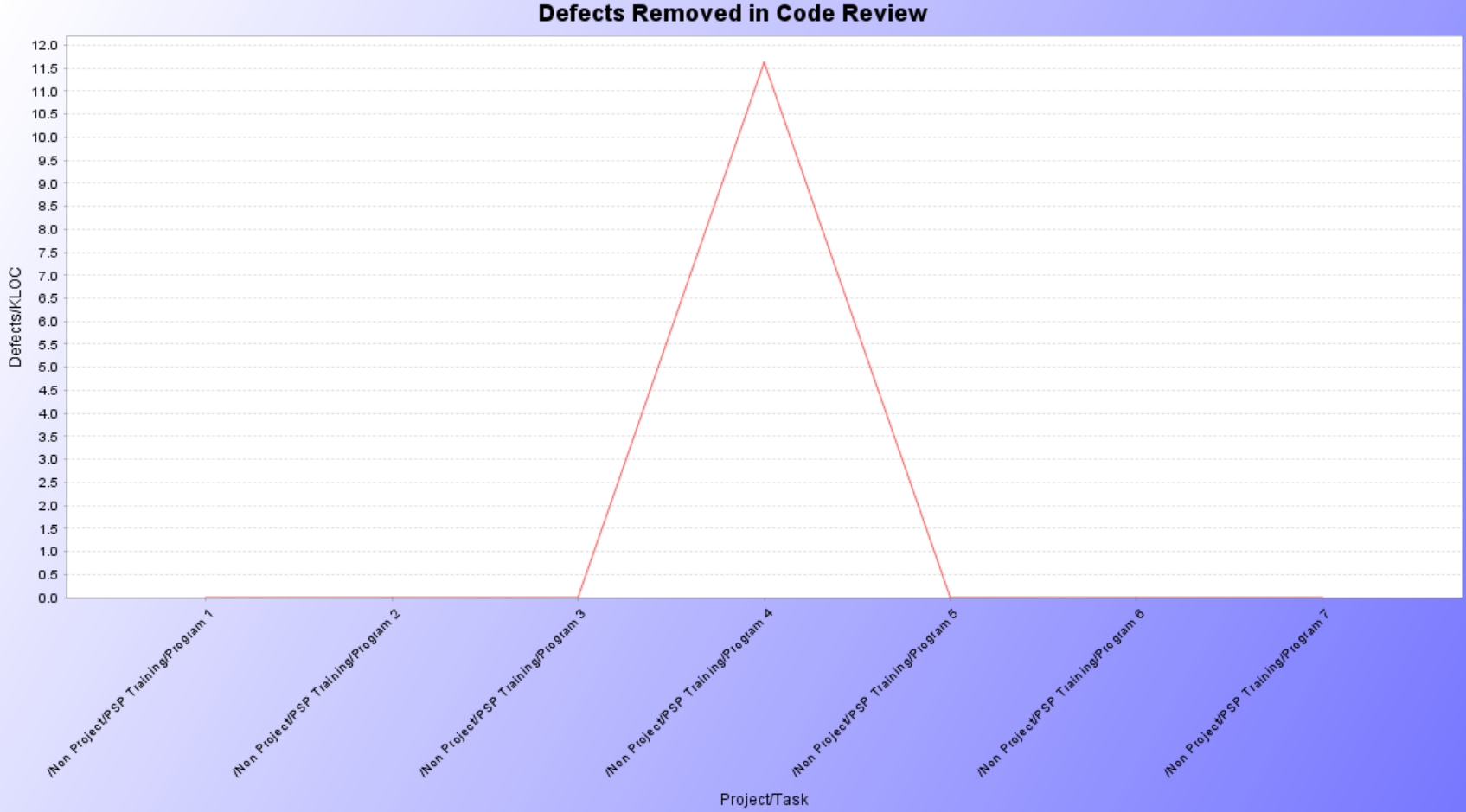


La tendencia ha sido a la baja desde el programa 1 hasta el 3, luego fue creciendo constantemente hasta alcanzar el pico en el programa 5 para luego empezar a decrecer abruptamente en el 6 y mantenerse estable en el 7

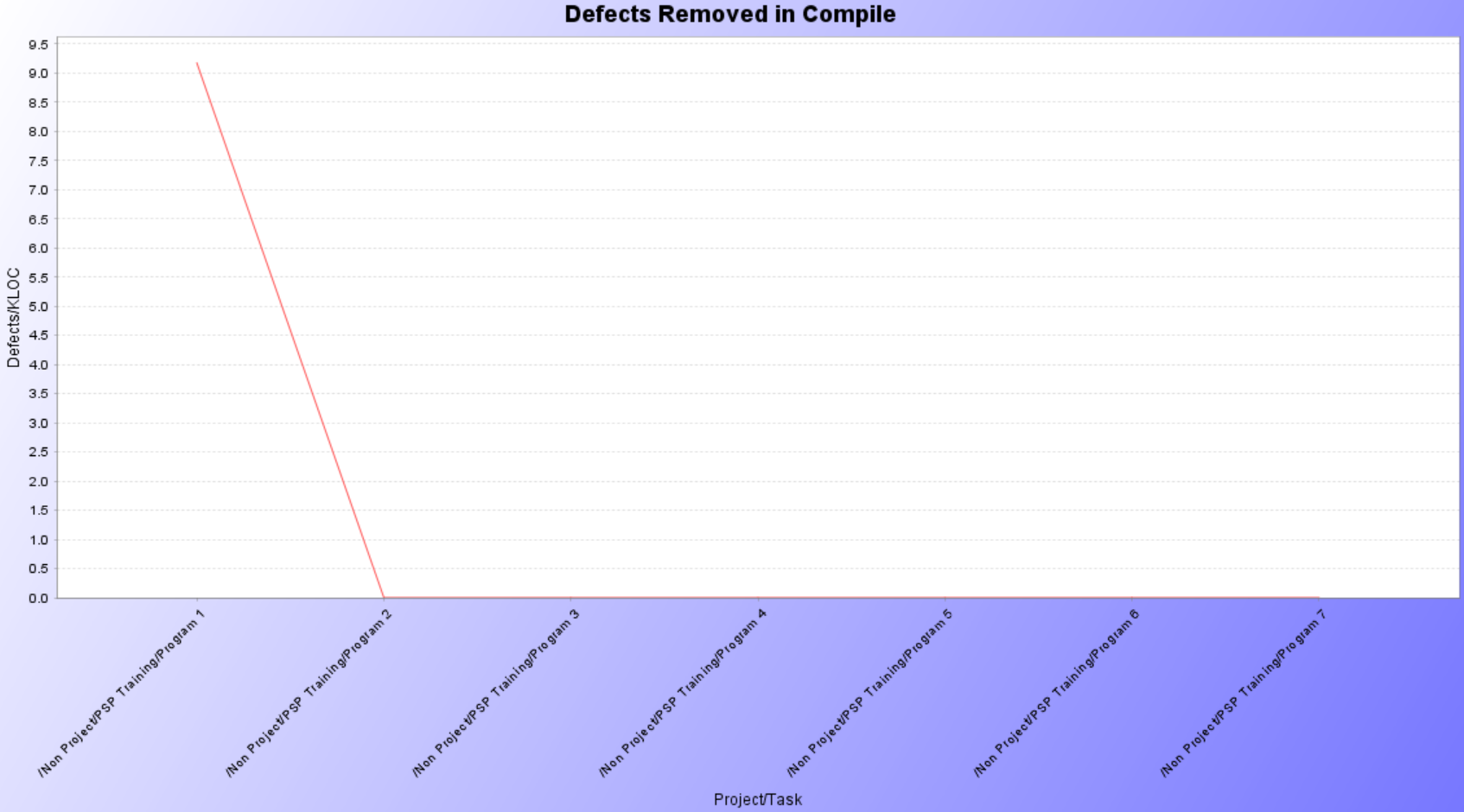
## ¿Cuál es la tendencia en los 7 programas de los defectos removidos por KLDC en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas?



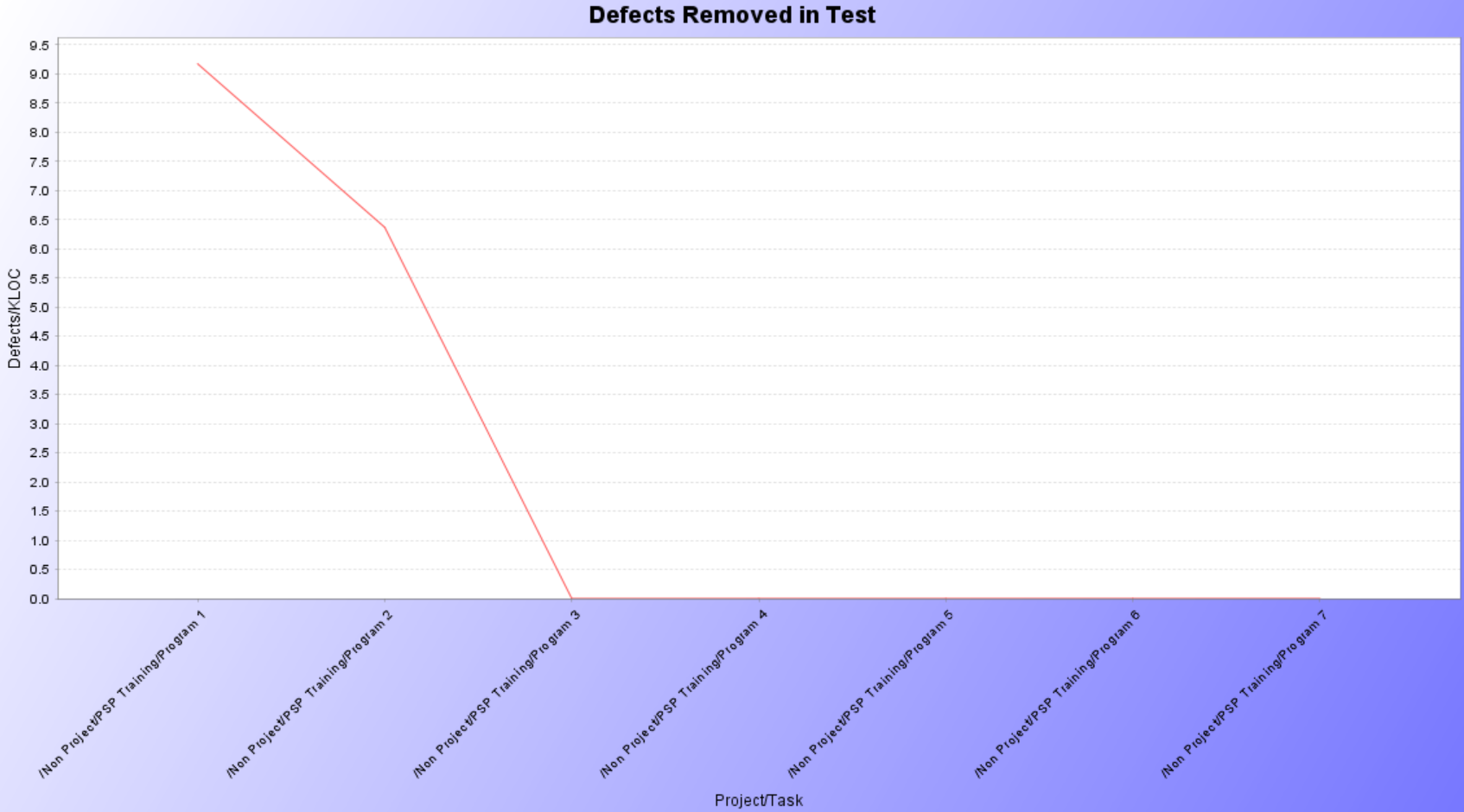
La tendencia fue que la primera vez usado elimino muchos errores llegando a tener 41.7 Defects/KLOC, de ahí ha bajado para mantenerse constante



La tendencia igual que la anterior grafica llego a tener su punto mas alto en 11.6 Defects/KLOC, para luego bajar y mantenerse en 0



Comenzó en un punto alto a 9.17 Defects/KLOC, para luego bajar a 0 en los laboratorios restantes



Comenzó en 9.17 Defects/KLOC para empezar a bajar y mantenerse en 0 a partir del laboratorio 3

## ¿Cómo se comparan las tasas de remoción de defectos (defectos removidos por hora) en las fases de Revisión del Diseño, Revisión del Código, Compilación y Pruebas en los 7 programas?

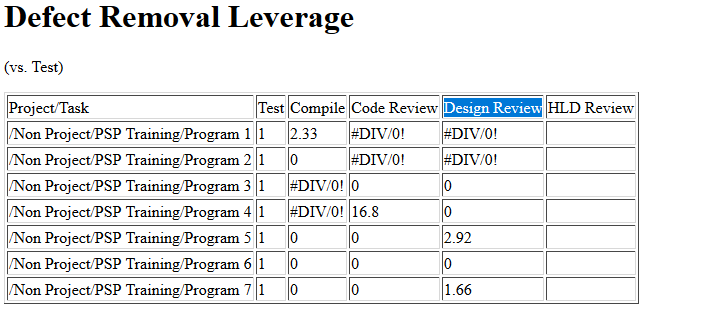
Que en las fases de revisión de diseño y codigo se logro remover muchos mas defectos que en las de compilación y pruebas, ya que la cantidad de defectos/KLOC son mayor en esas fases de revisión, esto ayuda a que los errores se eliminen lo mas antes posibles y no afecten el desarrollo del producto

## ¿Cuál fue tu velocidad (LDC por hora) a la que revisaste el código de cada programa?

* ***Programa 1***
  + Velocidad de Revision del codigo: ND
* ***Programa 2***
  + Velocidad de Revision del codigo: ND
* ***Programa 3***
  + Velocidad de Revision del codigo: 1960 LCD/Hr
* ***Programa 4***
  + Velocidad de Revision del codigo: 2790 LCD/Hr
* ***Programa 5***
  + Velocidad de Revision del codigo: 360 LCD/Hr
* ***Programa 6***
  + Velocidad de Revision del codigo: 1183 LCD/Hr
* ***Programa 7***
  + Velocidad de Revision del codigo: 740 LCD/Hr

## ¿Cuál fue la palanca de remoción de defectos (DRL) de cada programa para las fases de Revisión del Diseño y Revisión del Código? ¿Es bueno o malo?

* ***Programa 1***
  + Code Review: NA
  + Design Review: NA
* ***Programa 2***
  + Code Review: NA
  + Design Review: NA
* ***Programa 3***
  + Code Review: 0
  + Design Review: 0
* ***Programa 4***
  + Code Review: 16.8
  + Design Review: 0
* ***Programa 5***
  + Code Review: 0
  + Design Review: 2.92
* ***Programa 6***
  + Code Review: 0
  + Design Review: 0
* ***Programa 7***
  + Code Review: 0
  + Design Review: 1.66

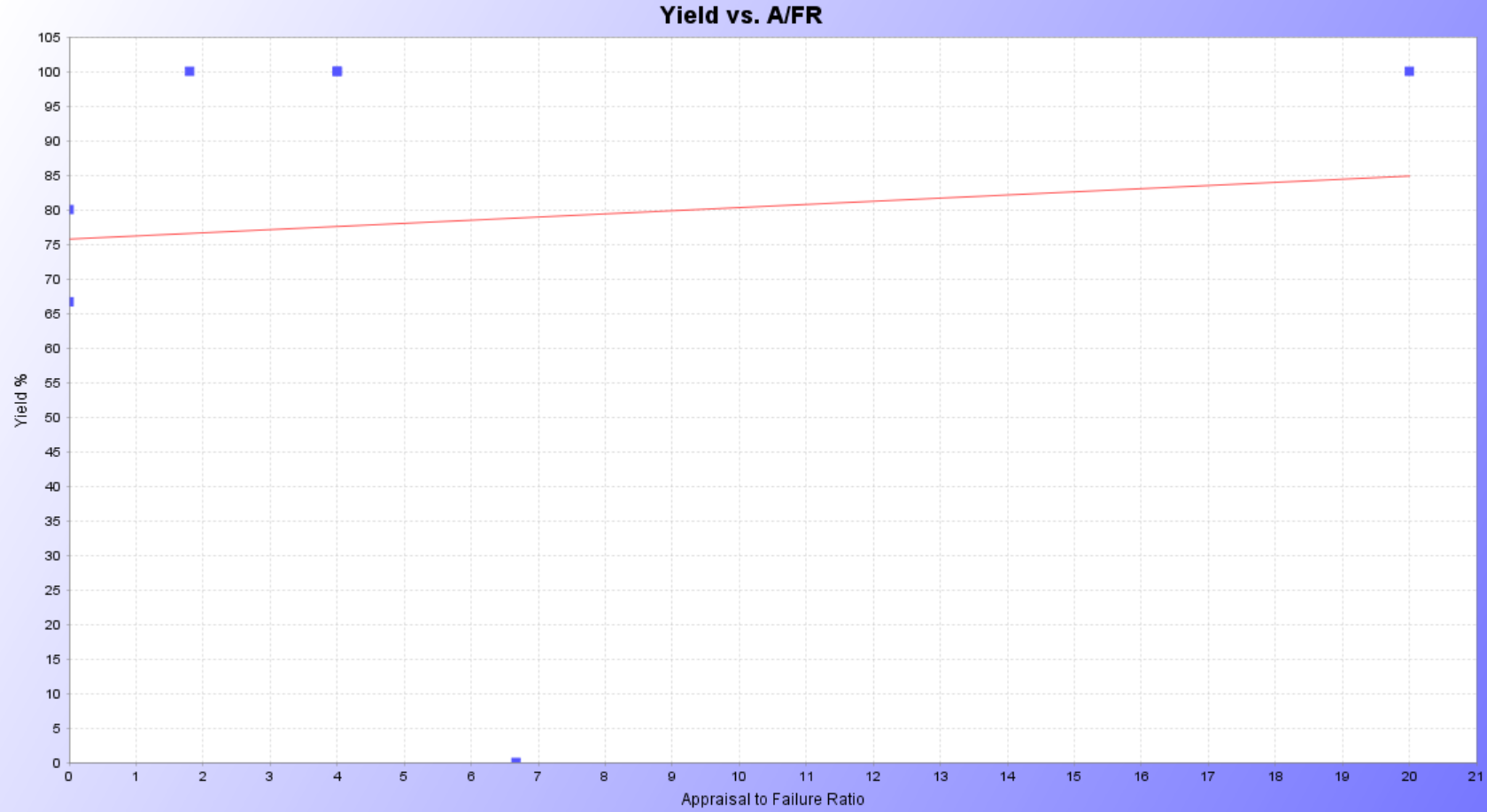


## ¿Existe alguna relación entre el Yield de cada programa y la velocidad (LDC por hora) a la que revisaste el código?

Si, ya que mientras mas alto era la velocidad a la que revisaba mas bajo era el yied

## ¿Existe alguna relación entre el Yield de cada programa y su A/FR?

Es directamente proporción a que mientras mas altas probabilidades haya de fracaso, tambien son las probabilidades de mayor rendimiento



## ¿Estás encontrando los defectos en las fases de Revisión del Diseño y del Código, o se te están escapando a las fases de Compilación y Pruebas? ¿Por qué?

Los estoy encontrando en la fase de revisión tanto de diseño como de codigo, ya que desde que se implementaron he revisado mis errores y me he dado cuenta a tiempo de los defectos que he encontrado, logrando asi evitar que lleguen a las fases de compilación y pruebas, esta técnica tambien me ha ayudado a evitar problemas en el desarrollo del codigo ya que los problemas se han evitado antes de que se presenten en la codificación de este mismo

## Basado en los datos históricos que has analizado sobre la calidad de tus programas, ¿Qué meta realista (alcanzable) te puedes poner para mejorar?

Mi meta realista seria la de seguir usando los procedimientos y empezar a hacerlos rutinas en mi codificación diaria, ya que he visto como la calidad de mi codigo ha ido mejorando con el pasar de los laboratorios, tambien seria la autoevaluación constante para saber que ando haciendo mal y cuales serían mis áreas de oportunidad

## ¿Qué cambios vas a hacerle a tu proceso personal para alcanzar esta meta?

Los cambios que hare serán mas que nada en abarcar mas errores que tuve y que no puse en su momento ya que no me habia dado cuenta, para poder asi disminuir la cantidad de errores que se me escapan y lograr tener un codigo mas limpios con menos errores

# Conclusión:

Como pudimos apreciar en lo presentado anteriormente, durante el transcurso de semestre se logró una notable mejoría en la calidad del software como tambien al momento de estimar el tamaño del software a desarrollar, de igual manera no solo se mejoraron las practicas si no que hubo una gran disminución en los fallos que el codigo iba presentando, comenzando en su punto mas altos para luego ir cayendo hasta llegar en una ocasión a tener 0 fallos en un codigo. De igual manera hubo ocasiones en las que se recaía a mismos fallos que previamente se tenía, por lo que cuando se añadieron las fases de revisión de diseño hubo una gran disminución de los errores, como se pudo apreciar en las gráficas.

Con los datos expuestos se da veracidad de que hubo un cambio sustancia al utilizar la técnica correspondiente, en la cual se inculco diferentes técnicas que sirven para la mejora personal y constante, ayudando asi a poder desarrollar un codigo de gran calidad desde el inicio de desarrollo de este mismo, logrando asi evitar entregar grandes cantidades de bugs al cliente

# Párrafo de honestidad:

Por medio de este párrafo yo Gerardo Aldair Ponce Gomez certifico que soy el autor intelectual del material que arriba entrego y que no realicé una copia de ninguna otra persona. En aquellos casos en los que tuve que incluir material de otra persona, reporté su apropiada referencia.