Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 25
Gestión de errores

IIC3745 – Testing

Rodrigo Saffie

rasaffie@uc.cl

2 de diciembre de 2020

Pruebas de sistemas

- *EC2 Mac instances*
- Probar aplicaciones en dispositivos Móviles: ¿Por dónde empezar?

Gestión de errores

- Aún cuando se realicen pruebas de manera exhaustiva pueden surgir errores en producción.
- Muchas veces los errores reportados por los usuarios son por mal uso del sistema.
- Es difícil solucionar errores si es que estos:
  - no son reportados ni detectados
  - no son replicables

Reporte de errores

- Se debe definir un canal y formato a través del cual se reportan los errores:
  - Número de telefono
  - Correo de soporte
  - Chat
    - *Zendesk*
    - *Intercom*
- Es importante filtrar, detectar duplicidad e impacto de estos errores.
- Si existe un error que está afectando a múltiples usuarios es útil comunicarlo para reducir los reportes: *status.io* / *GitHub status*

Reporte de errores

- Un buen reporte de error debería contener:
  - Resumen
  - Descripción
  - Pasos para reproducir
  - Comportamiento actual vs esperado
  - Ambiente de ejecución
  - Información adicional
- *Issues rubocop*

Replicar errores

- Algunos errores puede ser que solamente ocurran en producción.
- Los usuarios muchas veces no saben cómo ni qué describir de un error para ayudar a los desarrolladores.
- Es posible personificar usuarios para ver la aplicación exactamente como ellos:
  - *pretender*

Registrar errores: *logs*

- Es una buena práctica registrar los *logs* de las interacciones de los usuarios con el *software.*
- Los *logs* pueden entregar información relevante al momento de investigar causas de errores.
  - También pueden servir para obtener estadísticas sobre el uso.
- Existen distintos niveles de *logs*:
  - *Debug*
  - *Info*
  - *Warn*
  - *Error*
  - *Fatal*

Registrar errores: *logs*

- Para que un *log* sea útil debe:
  - Tener una estructura consistente
  - Proveer contexto de ejecución
  - Contener *timestamp* e identificador de acción
- Es peligroso que los *logs* registren información sensible de los usuarios
  - *Rails Log Filtering*
- Herramientas
  - *Papertrail*
  - *AWS CloudWatch*

Registrar errores: *tracking*

- Si bien los *logs* contienen la información relacionada a los errores, estos no son fáciles de analizar.
- Existen herramientas que automatizan el análisis de los *logs* para facilitar la trazabilidad de errores:
  - *Sentry*
  - *Raygun*
  - *Honeybadger*

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 25
Gestión de errores

IIC3745 – Testing

Rodrigo Saffie

rasaffie@uc.cl

2 de diciembre de 2020