

Nombre y apellido: Joaquín
Carbalini

Legajo:

Ingeniería de Software 2
LCC - FCEIA - UNR

Cristiá, Krapf, Scandolo
Lunes 18 de diciembre de 2023

Examen Final

Nota: La interpretación de las consignas es parte del examen. Cada parte se aprueba con al menos dos problemas bien y no menos del 65% del puntaje de esa parte. Problemas parcialmente correctos no necesariamente suman puntaje.

Primera parte

Duración aproximada 3 hs.

Se evalúa: Arquitecturas de software, estilos arquitectónicos, descripción de diseños en diferentes estilos: testing funcional y estructural.

1. Describa los módulos 2MIL, la guía de módulos y el diagrama canónico de un diseño basado en **Sistemas Estratificados** para los siguientes requerimientos.

El sistema permite al usuario colorear partes de figuras geométricas con diferentes colores. Las figuras geométricas son una simplificación de los objetos que se ven en una imagen tomada con una cámara de fotos digital. El sistema lee el archivo generado por la cámara y luego vectoriza la imagen (es decir, aproxima con curvas geométricas las secuencias de puntos de la imagen). El usuario puede indicar un color para el perímetro o superficie de una figura específica o para la intersección de varias de ellas. Cuando el usuario colorea parte de la imagen no solo lo ve en su pantalla sino que se modifica físicamente el archivo original.

2. Describa los módulos 2MIL, la guía de módulos, la configuración del **administrador de eventos** y la especialización seleccionada de un diseño basado en **Invocación Implícita** para los siguientes requerimientos.

Se desea contar con la posibilidad de registrar los eventos de seguridad que se producen al utilizar un sistema operativo tipo UNIX. Los eventos son emitidos por las **llamadas al sistema**. En general todas las llamadas emiten eventos que contienen la siguiente información: llamada al sistema, parámetros de la llamada, fecha y hora, proceso que invoca la llamada, usuario a nombre del cual ejecuta el proceso, valor de retorno de la llamada, etc.

La registración de eventos se debe hacer según lo que definan los administradores de la política de auditoría. Básicamente debe ser posible combinar de diversas formas los siguientes atributos: usuario, fecha, programa, tipo de operación, recurso sobre el cual se aplica la operación, etc.

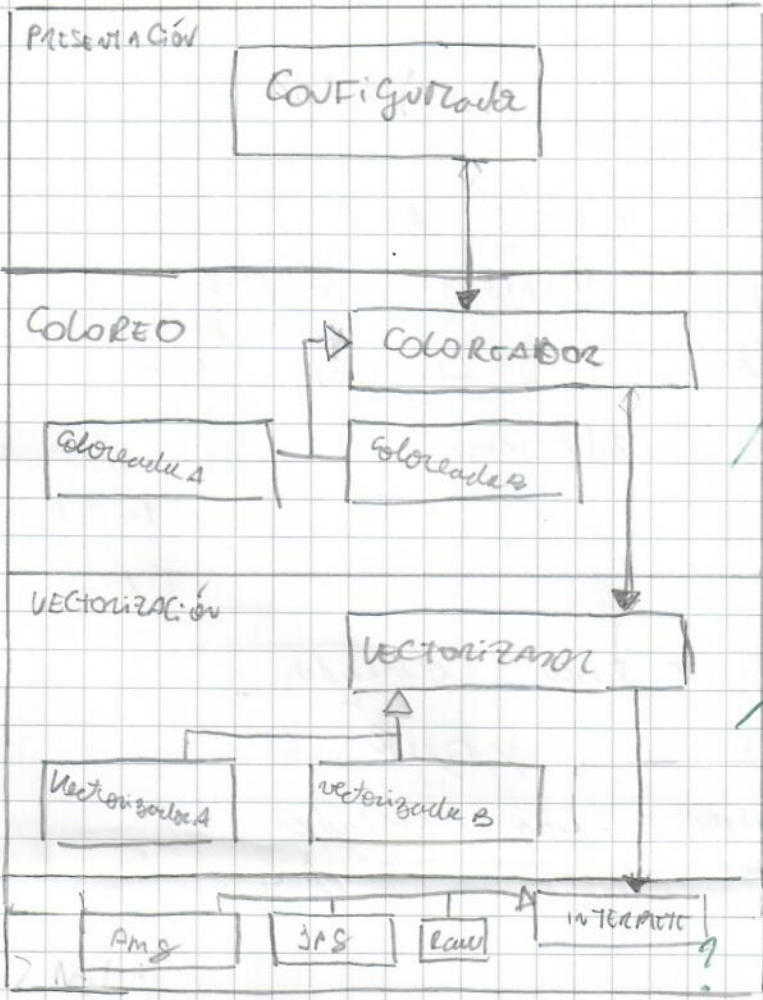
Definir en particular las siguientes políticas: auditar todos los intentos fallidos de apertura de cualquier archivo, auditar todos los archivos abiertos por el usuario root, auditar todos los accesos sobre el archivo /etc/passwd y auditar todos los eventos que se produzcan entre las 0:00 hs. y 6:00 hs.

3. Aplique el criterio de cubrimiento de flechas al siguiente fragmento de programa.

```
read(x);
max := x;
i := 2;
inic: if i > 100 then goto fin;
      read(x);
      if x > max then max := x;
      i := i + 1;
      goto inic;
fin: print(max);
```

4. Indique claramente los pasos del proceso de testing basado en especificaciones Z desde la generación de casos de prueba hasta la comprobación de los resultados.

1 Diagrama Canónico



SERVICIO: AMIGABLE Y FACIL USAR POR USUARIO

SERVICIO: Colocar una imagen

SERVICIO vectorizar una imagen

SERVICIO interactuar con los archivos y guardar las imagenes para trabajar.

```

MODULE Vectorizador
IMPORTS Imagen, Forma?
EXPORTS SetImagen (i String)
@Plicoz Vectorizado () :Imagen
setForma (i lista (Forma))

```

```

MODULE VectorizadoraA INHERITS FROM Vectorizador

```

```

MODULE VectorizadoraB INHERITS FROM Vectorizador

```

MODULE Coloredor

IMPORTS Imagen, Vectorizador, Color, Forma

EXPORTS ret color (i Color)

colorear Lineas (i Bool)

Colorear Relleno (i Bool)

Set imagen (i String) **DEBERIA SACARLO**

aplicar Coloredor (.): Imagen **DE UNA**

Set Forma (i List(Forma)) **LLAMADA**

A LA

CAPA INFERIOR

MODULE Coloredor A INHERITO FROM Coloredor

XQUIS

MODULE Coloredor B INHERITO FROM Coloredor

MODULE Configurador

IMPORTS Imagen, Coloredor, Color, Forma

EXPORTS ret color (i Color)

ret lineas y Relleno (i Bool, i Bool)

ret imagen (i string)

ret Forma (i List(Forma))

run ()

PRIVATE Guardar Archivo ()

Plot ()

```

MODULE Interpretate
Import Imagen
Export abrir(i String):Imagen

```

```

MODULE JPG INHERITE FROM Interpretate

```

```

MODULE PNG INHERITE FROM Interpretate

```

GUI

MODULO FORMAS

IMPORTS

EXPORTS

Güta de módulos:

1. Interpretar (estrato inferior)

propósito: abrir los distintos formatos de archivos de imágenes codificados de formas diversas

Secreto: como se interpretan los distintos tipos de archivos

2. Vectorizador

propósito: agregar a la imagen una nueva capa donde se encuentra el reconocimiento de las figuras medidas

Secreto: táctica de vectorizado

3. Coloreador

propósito: colorear las áreas en las capas de una imagen

Secreto: técnicas y tácticas de colores de formas

4. Presentación:

propósito interactuar con el usuario

Secreto: no exponer algoritmos y brindar interfaz amigable

5. Formas

propósito: abstraer el comportamiento de generas Formas en base a la unión e intersección de estas

2

Configuración del Administrador de Eventos

- * Los eventos se declaran en tiempo de compilación y están ligados 1 a 1 con los op codes de las sys calls
- * El evento anunciado es notificado a todos los suscriptores y estos deciden procesar si les es útil o no.
- * Cada evento tiene sus propios tipos fijos, que contemplan los administrativos (Unión, union, proceso, etc) como los útiles a cada llamada en particular
- * Los eventos se encuentran definidos todos juntos en un solo lugar.

Obs: el tod establece y que implementa las funcionalidades principales con los Sys Calls
los tody son los controladores

```
MODULE SysCall
IMPORTS OS, Pid, User, Date, InfoAdministrativa
EXPORTS
  retPid (i Pid)
  retUser (i User)
  retDate (i Date)
  sendInfo ( ) : InfoAdministrativa
```

```
MODULE Open INHERITS FROM SysCall
IMPORTS Fd, InfoAdministrativa
EXPORTS open (i String, i User) : Fd
ANNOUNCES open (InfoAdministrativa, Fd, String)
```

```
MODULE Close INHERITS FROM SysCall
IMPORTS InfoAdministrativa
EXPORTS close (i Fd) : Bool
ANNOUNCES close (InfoAdministrativa, Fd, Bool)
```

```

MODULE SyscallRegistrar
IMPORTS InfoAdministrativa, Fd,
EXPORTS OpenRegistrar( ... )
        CloseRegistrar( ... )
        ForkRegistrar( ... )
callson EVENT ?
// faltan el resto de SysCalls.

```

```

MODULE Intentos fallidos Alertar
INHERIT From SyscallRegistrar
Comments debe suscribirse al evento "open" el
resto de los interfaces no tienen
comportamientos

```

```

MODULE Archivos Abiertos
INHERIT From SyscallRegistrar
Comments debe suscribirse a los eventos
"open" y "close" notarse,
implementar interfaces para esos
eventos

```

```

MODULE Track Archivo
INHERIT From SyscallRegistrar
Comments implementar todas las interfaces
necesarias para ver como se
comprueba el archivo "/etc/passwd"

```

```

MODULO Adm Eventos
IMPORT SysCall registrar, ...

EXPORTS
  Suscribir Oren (i SysCall registrar)
  Anunciar Oren (Info Administrativa, Fd, String)
  Suscribir Close (i SysCall registrar)
  Anunciar Close (Info Administrativa, Fd, Bool)

```

GUIA de Módulos

1. Adm Eventos

Secretos: Ocultar la petición de del administrador

Propósito: Manejar el número de eventos y la invocación de aquellos que lo esperaban

2. SysCall (módulo Abstracto)

Propósito: Unificar la forma que tienen todas las Sys Calls de generar la información administrativa

2.1 Oren

Secretos: Como se abre un archivo

Propósito: Permite a procesos abrir archivos, además genera un evento cada vez que es invocada.

3.3 Troce Archivo.

Proposito: Nos involucra con los eventos que interactúan con el archivo.

Secreto: idem que anterior

3.4 Archivos root

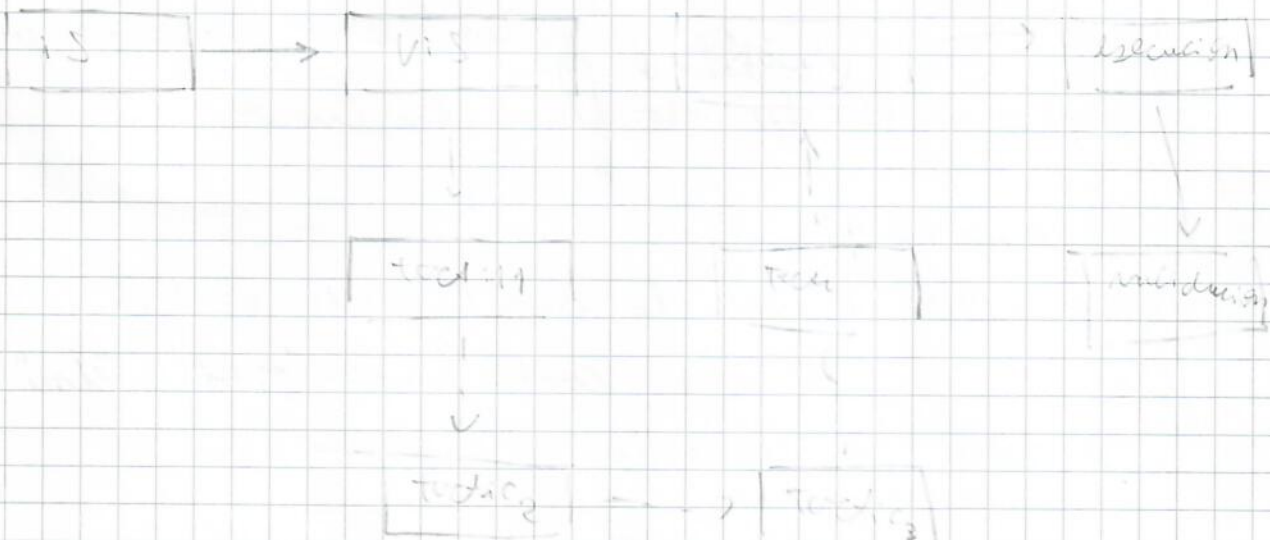
Proposito se invoca cada vez que el root interactúa con un archivo, a través de los hooks relacionados al manejo de Archivos.

Secreto: idem anterior

3.5 Zona horario Protegida

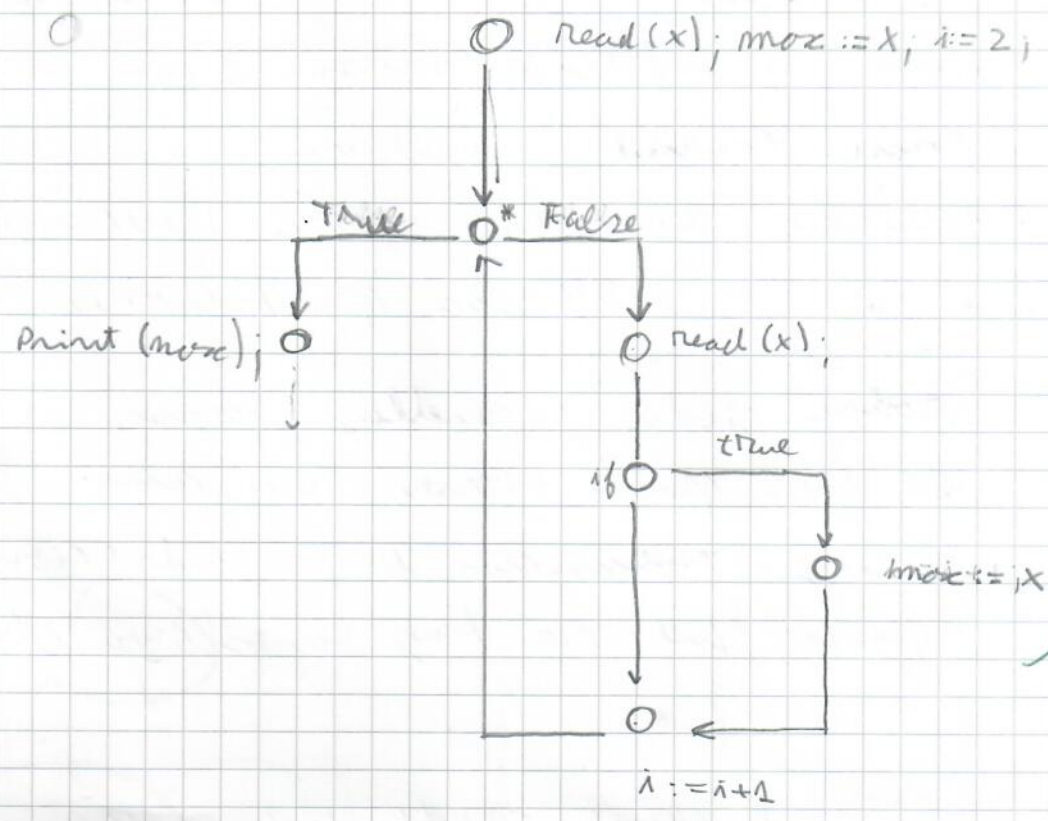
Proposito: a través de los eventos de los SysCalls se registran todos los eventos ocurridos

Secreto: idem anterior.



3

Grafo



* se reescribe el bucle a un formato conocido, similar al del while, pero con las condiciones al revés

La estrategia necesaria para lograr un cubrimiento de flechas es por ejemplo introducir un 2, luego un un 3 y después repetidamente 3 otras 98 veces. Con esta base de prueba se recorre al menos una vez cada una de las flechas del grafo

$\# = \{ (2, \underbrace{3, 3, \dots, 3}_{99 \text{ veces}}) \}$ este $\#$ es el mínimo que recorre todas las flechas.

4

1. dar todo el input space (iS)
2. restringirlo a solo los validos (v.i)
3. llevar la especificación 2 a Formas normales de conjuntos
4. Aplicar taticas para restringir el viS a diversas particiones
5. probar todas a aquellas ramas de las particiones que son vacias (terminan en contradicciones, por lo que no hay casos reccios reccios?)
6. Generar el conjunto de soluciones para cada particion generada
7. evaluar

