Laboratorio 4: Paradigma Orientado a eventos

Rodrigo Mardones Aguilar

9/6/2020

Indice

Introducción

Los Paradigmas de programación son modelos o patrones de trabajo, los cuales nos permiten resolver problemas abordando de distintas manera la ejecución de la solución.

Es preciso señalar que indiferente de los lenguajes nuevos, modernos y sus herramnientas propias asociadas, siempre se regirán por al menos uno o más paradigmas de programación. Pues son estos quienes nos permiten analizar de buena manera.

Objetivos generales

Para este laboratorio se plantea crear una pieza de software dentro del paradigma de programación orientado a eventos que tome lo desarrollado en el informe anterior para realizar una interfaz que permita interactuar con la simulación de un "controlador de versiones" como lo es GIT.

Objetivos especificos

Para abordar el problema se plantean lo siguientes objectivos especificos

- Diseño de la solución.
- Aspectos de la implementación.
- Instrucciones de uso.
- Resultados y Autoevalución.
- Conclusiones.

Descripción del problema

Descripción del problema

La necesidad de mantener y producir software escalable, ordenado y mantenible durante el tiempo es una cuestión importante dentro de la industria. Poder llevar todos los cambios asociados a un proyecto, trabajado por un grupo importante de personas y todo lo que conlleva el manejo de errores es parte de lo que resuelven los software denominados “controladores de versiones”.

Para el problema en cuestión se requiere de una pieza de software que permita realizar las siguientes acciones dentro de un directorio de trabajo, independiente del trabajo realizado dentro de este:

- implementar una interfaz de usuario que permita interactuar con los distintos estados de git
- iniciar un repositorio desde cero.
- ver los status de cada zone de trabajo con botones que accedan a estos.
- realizar cada acción pertinente a los laboratorios de git anteriores (init, add, commit, push, pull, agregar archivo).

Descripción del paradigma

El "paradigma orientado a eventos" es un paradigma que se centra (como su nombre lo indica) en la detección de eventos. El "evento" es la unidad principal de este y se entiente por evento a una acción o ocurrencia que el programa puede detectar y manejar a anotojo. Estos eventos pueden disparar otras acciones ya sean sincronas o asincronas dentro de la ejecución de nuestro programa.

Para poder diferenciar mejor las entidades que componen a este paradigma podemos diferencias los siguientes puntos:

- Evento : Unidad principal dentro del paradigma, es la acción o reacción de un usuario o maquina. la cual dispara el evento en cuestión.
- manejador de eventos: Es una subrutina que se ejecuta tras la llamada de un evento especifico. Esta subrutina es un trozo de codigo que toma el evento y lo maneja, realizando una acción especifica asociada al evento, Ejemplo(click de un boton, lectura de un sensor).
- Delegado: Los delegados son estructuras que podemos apreciar en algunos lenguajes y que hacen referencia a nivel de memoria a una funcion que se ocupa como un "callback" o para realizar eventos personalizados.
- Excepción: las excepciones son estructuras de codigo que nos permiten manejar posibles errores que puedan ocurrir en tiempo de ejecución al momento de manejar eventos o subrutinas asociados a eventos. Estas excepciones pueden ser manejadas y aisladas para entregar errores entendibles y poder continuar con flujos de trabajo sin mayor impacto de cara al usuario.

En general los lenguajes que implementan este paradigma poseen eventos ya predefinidos, generalemente asociados a acciones realizadas por el usuario, sistema o el hardware donde es utilizado el programa en cuestión.

Análisis del problema

Diseño de la solución

Aspectos de la implementación

Instrucciones de uso

Resultados y Autoevaluación

Conclusiones

Referencias