Christopher Zúñiga C28730
Mario Cordero C22306

Especificación de Funcionalidad de la Aplicación

Descripción General:

La aplicación propuesta analizará un archivo de log complejo que simula registros generados por sistemas operativos o aplicaciones reales, permitiendo a los usuarios interactuar con esta información mediante una interfaz gráfica amigable y funcional.

Opciones del Usuario:

La aplicación contará con al menos tres funcionalidades principales:

1. **Visualización y Búsqueda de Eventos**
   - Descripción: Permite al usuario ver todas las entradas de log y realizar búsquedas filtradas por:
     - Fecha: Rango de fechas específico.
     - Nivel de Log: INFO, WARN, DEBUG, ERROR.
     - Palabras Clave: Búsqueda de mensajes específicos.
   - Interfaz:
     - Una tabla central que muestra las entradas de log filtradas.
     - Un panel lateral con opciones de filtrado.
     - Doble clic en una fila para ver detalles completos (incluyendo details de LogEntry).

---

2. **Análisis de Reportes de Fallo (Crash Reports)**
   - Descripción: Presenta informes detallados de los fallos del sistema, mostrando:
     - Código de Error: Identificador único del fallo.
     - Mensaje de Error: Descripción del fallo.
     - Patrones Frecuentes: Identificación de fallos recurrentes.
   - Interfaz:
     - Un gráfico de barras que muestra la frecuencia de fallos por tipo de error.

Christopher Zúñiga C28730
Mario Cordero C22306

3. Visualización de Estadísticas de Copias de Seguridad

- Descripción: Ofrece un gráfico interactivo que muestra:
    - Progreso Promedio: Tiempo y porcentaje de completitud de las copias de seguridad.
    - Frecuencia: Número de copias de seguridad realizadas en un período.
    - Duración: Tiempo total de las copias de seguridad.
- Interfaz:
    - Un gráfico de líneas que muestra el progreso de las copias de seguridad a lo largo del tiempo.
    - Un resumen de estadísticas clave (tiempo promedio, frecuencia, tasa de éxito/fallo).
    - Una lista de copias de seguridad con detalles como archivos incluidos y actualizaciones de progreso (BackupUpdate).

Interfaz Gráfica (GUI):

Diseño General

- **Herramienta**: La interfaz gráfica se desarrollará utilizando PyQt5 en Python.
- **Estilo**: Interfaz limpia y moderna, con colores neutros (blanco, gris, azul) para el fondo y colores llamativos (rojo, verde, amarillo) para resaltar información importante.

Estructura de la Interfaz

1. Barra de Menú Superior

- **Inicio**: Para volver a la pantalla principal.
- **Archivo**: Opciones para cargar un archivo de log, guardar resultados o exportar datos.
- **Herramientas**: Acceso rápido a las funcionalidades principales (visualización de eventos, análisis de fallos, estadísticas de copias de seguridad).
- **Ayuda**: Información sobre cómo usar la aplicación y detalles del proyecto.

2. Panel Principal

El panel principal tendrá pestañas o secciones para cada funcionalidad:

a. Visualización y Búsqueda de Eventos

- **Filtros:**
  - Selector de rango de fechas.
  - Checkboxes para niveles de log (INFO, WARN, DEBUG, ERROR).
  - Campo de texto para buscar palabras clave.

- **Tabla de Eventos:**
  - Columnas: Timestamp, Nivel, Mensaje.
  - Doble clic en una fila para ver detalles completos.

b. Análisis de Reportes de Fallo (Crash Reports)

- **Gráfico de Frecuencia de Fallos:**
  - Gráfico de barras que muestra la frecuencia de fallos por tipo de error.

- **Lista de Fallos:**
    - Lista de fallos con opción de expandir para ver detalles (código de error, mensaje, traza de la pila).

c. Visualización de Estadísticas de Copias de Seguridad

- **Gráfico de Progreso:**
    - Gráfico de líneas que muestra el progreso de las copias de seguridad a lo largo del tiempo.
- **Resumen de Copias de Seguridad:**
    - Estadísticas clave (tiempo promedio, frecuencia, tasa de éxito/fallo).
- **Detalles de Copias:**
    - Lista de copias de seguridad con detalles como archivos incluidos y actualizaciones de progreso.

3. Barra Inferior (Status Bar)

- Muestra información útil como:
    - Estado de la aplicación (por ejemplo, "Archivo cargado correctamente").
    - Número de eventos cargados.
    - Tiempo de última actualización.

Gráficos Interactivos

- Herramientas:
    - Matplotlib: Para gráficos estáticos (barras, líneas, tortas).
    - Plotly: Para gráficos interactivos (zoom, filtros, detalles al hacer clic).
- Ejemplos de Gráficos:
    - Gráfico de barras para la distribución de niveles de log.
    - Gráfico de líneas para el progreso de copias de seguridad.
    - Gráfico de torta para el porcentaje de fallos vs. operaciones exitosas.

Christopher Zúñiga C28730
Mario Cordero C22306

Estructuras de Datos Específicas:

Clase LogEntry

```
class LogEntry:
    timestamp: datetime     # Sello de tiempo de la entrada de log.
    level: str              # Nivel de severidad (INFO, WARN, DEBUG, ERROR).
    message: str            # Mensaje breve asociado a la entrada.
    details: Optional[Union[Diagnostic, BootSequence, Backup, CrashReport]]# Detalles adicionales.
```

Clase Diagnostic

```
class Diagnostic:
    checks: Dict[str, Dict[str, str]]   # Diccionario de comprobaciones diagnósticas.
                                        # Ejemplo: {'video': {'result': 'pass', 'latency': '28ms'}}
```

Clase BootSequence

```
class BootSequence:
    steps: List[str]  # Lista de pasos en la secuencia de arranque.
```

Clase Backup

```
class BackupUpdate:
    timestamp: str  # Sello de tiempo de la actualización.
    progress: int   # Progreso de la copia de seguridad (en porcentaje).
    details: str    # Detalles adicionales de la actualización.

class Backup:
    source: str             # Fuente de la copia de seguridad.
    destination: str        # Destino de la copia de seguridad.
    file_list: List[str]    # Lista de archivos incluidos en la copia.
    updates: List[BackupUpdate]  # Lista de actualizaciones de progreso.
```

Tarea Progamada 1
Entregable#1

Christopher Zúñiga C28730
Mario Cordero C22306

Clase CrashReport

```
class StackFrame:
    function: str  # Nombre de la función donde ocurrió el error.
    line: int      # Número de línea donde ocurrió el error.

class CrashReport:
    error_code: int         # Código de error asociado al fallo.
    message: str            # Mensaje descriptivo del fallo.
    stack_trace: List[StackFrame]  # Traza de la pila (stack trace) del fallo.
```

Christopher Zúñiga C28730
Mario Cordero C22306

Robustez y Validación:

- El analizador léxico identificará correctamente tipos de tokens como timestamps, niveles de log, palabras clave, estructuras anidadas, arreglos y objetos.
- El analizador sintáctico asegurará que la estructura del archivo sea correcta, validando la jerarquía y los tipos de datos esperados, generando alertas claras sobre errores en la estructura o tipos de datos incorrectos.