

Diseño de Pruebas para el Módulo de Depuración

Objetivo del Plan de Pruebas

El objetivo es validar que las funciones de depuración, como la detección de duplicados, la corrección de salidas faltantes, la corrección de marcas invertidas, y la depuración del archivo, funcionen de acuerdo con los requisitos especificados. Se debe asegurar que los datos sean procesados correctamente y que las salidas sean las esperadas.

Alcance de las Pruebas

El plan cubre las siguientes funcionalidades del módulo:

1. **Detección de duplicados:** Validar que los registros duplicados sean correctamente identificados y etiquetados.
2. **Corrección de salidas faltantes:** Asegurarse de que las salidas que falten sean corregidas correctamente.
3. **Corrección de marcas invertidas:** Verificar que las marcas de entrada/salida invertidas se corrijan adecuadamente.
4. **Depuración de archivo:** Validar que el archivo de entrada se procese correctamente, devolviendo un DataFrame o None según el caso.

Casos de prueba

| ID Prueba | Caso de Prueba | Entrada | Resultado Esperado | Estado |
|-----------|-------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------|
| PDD-001 | Detección de duplicados | Datos de prueba con registros duplicados. | La función duplicados() debe identificar los duplicados en los registros de entrada. | Passed |
| PDD-002 | Validar que las salidas faltantes sean correctamente corregidas | Registro sin hora de salida | La función faltaSalida() debe corregir las salidas faltantes basándose en las reglas de horario. | Passed |
| PDD-003 | Asegurarse de que las marcas invertidas de entrada y salida se corrijan | Datos con marcas invertidas en la columna "entrada/salida" | La función marcaOpuesto() debe corregir los valores invertidos de las marcas de entrada/salida. | Passed |
| PDD-004 | Validar que el archivo de entrada sea procesado correctamente. | Un archivo .log con datos de prueba. | El resultado debe ser None o un DataFrame | Passed |

Criterios de Aceptación

- **Duplicados:** La función debe identificar correctamente los registros duplicados y etiquetarlos adecuadamente con la columna "Error".
- **Salidas faltantes:** Debe corregir las salidas faltantes y crear nuevas salidas con la hora correspondiente si es necesario.
- **Marcas invertidas:** Debe corregir las marcas invertidas de entrada y salida de acuerdo con las reglas definidas.
- **Depuración de archivo:** La función debe manejar archivos correctamente, devolviendo un DataFrame procesado o None en caso de que ocurra un error.

Diseño de Prueba para la función de historial

Objetivo del Plan de Pruebas

El módulo `crearHistorial` es responsable de generar un historial de registros a partir de los datos procesados y almacenarlos en un archivo CSV. Las pruebas unitarias aseguran que esta funcionalidad cumpla con los requisitos, incluyendo la lectura de usuarios, manejo de errores y escritura correcta de los datos.

Alcance de las pruebas

El plan cubre las siguientes funcionalidades del módulo:

- 1. Lectura de `username.txt` para identificar al usuario
- 2. Escritura del archivo CSV con los datos del historial.
- 3. Manejo de errores al leer el archivo de configuración.
- 4. Verificación de la estructura y contenido del CSV generado.

Casos de prueba

| ID Prueba | Caso de Prueba | Entrada | Resultados Esperados | Estado |
|-----------|-----------------------------------------|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------|
| HST-001 | Lectura de <code>username.txt</code> | Archivo <code>username.txt</code> con el contenido <code>test_user</code> | La función lee correctamente el nombre del usuario y lo utiliza en el CSV generado. | Passed |
| HST-002 | Escritura del historial csv | DataFrame de prueba | El archivo <code>historial.csv</code> se guarda con el formato y datos correctos. | Passed |
| HST-003 | Manejo de ausencia de índices | DataFrame de prueba y lista vacía de índices | El archivo CSV se genera sin errores. | Passed |
| HST-004 | error al leer <code>username.txt</code> | Archivo <code>username.txt</code> Ausente o innaccesible | La función no intenta escribir el archivo CSV y no genera excepciones inesperadas. | Passed |

Criterios de Aceptación

- **Lectura de `username.txt`:** El archivo se lee correctamente y el nombre de usuario se incluye en el historial.
- **Escritura del archivo CSV:** El archivo `historial.csv` contiene las columnas `usuario`, `rut`, `fecha`, `error` y `cambio` con los datos correctos.

- **Manejo de Errores:** La ausencia de `username.txt` no genera excepciones y evita la escritura del archivo.
- **Ejecución sin índices:** La función maneja listas vacías de índices sin errores.

Diseño de Prueba para la función de Carga de Archivos

Objetivo del Plan de Pruebas

Evaluar la funcionalidad del módulo de subida de archivos implementado en Flask. El objetivo principal es validar que las características y escenarios de carga, validación y procesamiento de archivos funcionen correctamente, asegurando un flujo robusto y sin errores.

Alcance de las pruebas

El plan cubre las siguientes funcionalidades del módulo:

1. Subida de archivos con extensiones válidas.
2. Validación de errores al no adjuntar archivos.
3. Validación de errores al adjuntar un archivo sin nombre
4. Validación de extensiones no permitidas.
5. Verificación del almacenamiento correcto del nombre original del archivo.
6. Transformación del archivo subido a formato `.csv`.

| ID | Caso de Prueba | Entrada | Resultados Esperados | Estado |
|---------|--------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| PSA-001 | Subir un archivo válido <code>.log</code> | Archivo: <code>registro.log</code> | - El archivo se guarda en la ruta especificada. - El sistema redirige a <code>/visualizacion</code> tras la depuración. | Passed |
| PSA-002 | No enviar ningún archivo | Ningún archivo | - El sistema muestra un mensaje de error: "No se detecta archivo". - Redirige a <code>/carga</code> . | Passed |
| PSA-003 | Subir un archivo con nombre vacío | Archivo sin nombre | - El sistema muestra un mensaje de error: "No se detecta archivo". - Redirige a <code>/carga</code> . | Passed |
| PSA-004 | Subir un archivo con extensión no permitida | Archivo: <code>archivo.txt</code> | - El sistema muestra un mensaje de error: "Tipo de archivo inválido, intente nuevamente con un archivo tipo <code>.log</code> ". - Redirige a <code>/cargar</code> . | Passed |
| PSA-005 | Verificar que se guarda el nombre original del archivo | Archivo: <code>registro.log</code> | - El archivo <code>NOMBRE_ORIGINAL_ARCHIVO.txt</code> Contiene el nombre <code>registro.log</code> . | Passed |
| PSA-006 | Cambiar extensión del archivo a <code>.csv</code> | Archivo: <code>registro.log</code> | - El archivo guardado tiene la extensión <code>.csv</code> . | Passed |

| | | | | |
|--|---------------|--|--|--|
| | tras la carga | | | |
|--|---------------|--|--|--|

Criterios de Aceptación

- El módulo debe aceptar y procesar únicamente archivos con extensión **.log**.
- Los mensajes de error deben ser claros y específicos para cada tipo de fallo.
- Los archivos válidos deben ser guardados correctamente en la ubicación especificada.
- La función **depurar_archivo** debe ejecutarse solo si el archivo cumple con los requisitos iniciales.
- Tras un procesamiento exitoso, el sistema debe redirigir a **/visualizacion**.

Diseño de Pruebas para el Módulo de Validación

Objetivo del Plan de Pruebas

Garantizar que el módulo de validación funcione conforme a los requisitos especificados, asegurando la identificación y corrección precisa de errores en los registros procesados.

Alcance de las Pruebas

El plan abarca las siguientes funcionalidades del módulo:

- Validación de entradas duplicadas.
- Validación de corrección automática de salidas.
- Manejo de entradas invertidas a salida y viceversa.
- Verificación de la ejecución de la función `crearHistorial`.

Casos de pruebas

| ID | Caso de prueba | Entrada | Resultado esperado | Estado |
|---------|---------------------------------------------------------------|-----------------------------------|---------------------------------------------------------------------------------------------------|--------|
| PVD-001 | Validar entrada duplicada | Datos corregidos y fila con error | El error "Entrada duplicada" no debe estar presente en el resultado. | Passed |
| PVD-002 | Validar salida automática corregida | Datos corregidos y fila con error | El índice relacionado con el error corregido no debe estar presente en el resultado. | Passed |
| PVD-003 | Validar salida invertida a entrada | Datos corregidos y fila con error | El valor de "entrada/salida" debe ser igual a 3 para el índice correspondiente. | Passed |
| PVD-004 | Validar entrada invertida a salida | Datos corregidos y fila con error | El valor de "entrada/salida" debe ser igual a 1 para el índice correspondiente. | Passed |
| PVD-005 | Verificar que se llama la función <code>crearHistorial</code> | Datos corregidos y fila con error | La función <code>crearHistorial</code> debe ser llamada una vez durante el proceso de validación. | Passed |

Criterios de Aceptación

- **Entrada duplicada:** El sistema debe asegurar que no se registren entradas duplicadas en el DataFrame resultante.
- **Salida automática corregida:** Las salidas que se corrijan automáticamente deben ser eliminadas del resultado.
- **Entradas y salidas invertidas:** Los errores de entradas invertidas a salida o viceversa deben ser corregidos manteniendo la consistencia de los valores de "entrada/salida".
- **Ejecución de `crearHistorial`:** La función `crearHistorial` debe ser llamada con los índices procesados para generar el historial.

Diseño de Pruebas para el Módulo de Visualización

Objetivo del Plan de Pruebas

Validar que el módulo de visualización, edición y validación de datos permita a los usuarios interactuar con los registros procesados de manera eficiente, segura y precisa, garantizando que todas las funcionalidades operen según las especificaciones.

Alcance de las Pruebas

El plan de pruebas abarca las siguientes funcionalidades del módulo:

1. **Visualización de datos:** Verificar que la página de visualización se carga correctamente.
2. **Aplicación de filtros:** Asegurarse de que los filtros sean aplicados correctamente en las solicitudes.
3. **Descarga de archivos CSV:** Validar que la funcionalidad de descarga de archivos CSV funciona como se espera.
4. **Acceso a la visualización sin archivo disponible:** Verificar que el sistema redirige correctamente si el archivo necesario no está disponible.
5. **Acceso no autenticado:** Validar que los usuarios no autenticados sean redirigidos correctamente al intentar acceder a la visualización.

Casos de pruebas

| ID Prueba | Caso de Prueba | Entrada | Resultados Esperados | Estado |
|-----------|--------------------------------------------------------|----------------------------------------------|-------------------------------------------------------------------------------------|--------|
| PDV-001 | Visualizar la página de visualización | Solicitud GET a <code>/visualizacion</code> | El código de estado es 200, y la respuesta contiene "Visualización". | Passed |
| PDV-002 | Aplicar filtros | Solicitud POST a <code>/apply_filters</code> | El código de estado es 200, y la respuesta contiene "Filtros aplicados". | Passed |
| PDV-003 | Descargar archivo CSV | Solicitud POST a <code>/download_csv</code> | El código de estado es 200, y la respuesta contiene "Descarga CSV". | Passed |
| PDV-004 | Verificar redirección si el archivo no está disponible | Solicitud GET a <code>/visualizacion</code> | El código de estado es 302, y la ubicación de redirección es <code>/cargar</code> . | Passed |
| PDV-005 | Verificar redirección si el acceso es no autenticado | Solicitud GET a <code>/visualizacion</code> | El código de estado es 302, y la ubicación de redirección es <code>/login</code> . | Passed |

Criterios de Aceptación

1. **Visualización de Datos:** La página de visualización debe cargarse correctamente con un código de estado 200 y el mensaje adecuado en la respuesta.
2. **Aplicación de Filtros:** Los filtros aplicados deben reflejarse correctamente en la respuesta del servidor.
3. **Descarga de CSV:** El archivo CSV debe descargarse correctamente sin errores de servidor.
4. **Archivo No Disponible:** Si el archivo no está disponible, la aplicación debe redirigir al usuario a la página [/cargar](#) con un código de estado 302.
5. **Acceso No Autenticado:** Si el usuario no está autenticado, debe ser redirigido al login con un código de estado 302.