**📚 Documentación de Estructura y Variables - Baseline Adjustment Tool**

**📁 Estructura de Archivos**

baseline\_adjustment\_tool/

│

├── app.py # Router principal de la aplicación

├── config.py # Constantes y configuración

├── session\_manager.py # Gestión de st.session\_state

├── requirements.txt # Dependencias Python

│

├── ui/ # Módulos de interfaz de usuario

│ ├── \_\_init\_\_.py

│ ├── sidebar.py # Barra lateral con progreso

│ ├── step\_00\_client\_info.py # Paso -1: Datos del cliente

│ ├── step\_01\_backup.py # Paso 0: Advertencia backup

│ ├── step\_02\_wstd.py # Paso 1: Diagnóstico WSTD

│ ├── step\_03\_kit.py # Paso 2: Standard Kit (2 archivos TSV)

│ ├── step\_04\_correction.py # Paso 3: Cálculo de corrección

│ ├── step\_05\_baseline.py # Paso 4: Cargar baseline

│ ├── step\_06\_export.py # Paso 5: Exportar resultados

│ └── utilities.py # Utilidades (conversión archivos)

│

├── core/ # Lógica de negocio

│ ├── \_\_init\_\_.py

│ ├── file\_handlers.py # Lectura/escritura de archivos

│ ├── spectral\_processing.py # Procesamiento espectral NIR

│ └── report\_generator.py # Generación de informes HTML

│

└── utils/ # Utilidades auxiliares

├── \_\_init\_\_.py

├── validators.py # Funciones de validación

└── plotting.py # Funciones de graficación (Plotly)

**🗂️ Variables de Session State**

**Estructura Completa de st.session\_state**

python

st.session\_state = {

*# ============================================*

*# CONTROL DE FLUJO*

*# ============================================*

'step': int, *# Paso actual (-1 a 5)*

'backup\_done': bool, *# Si confirmó backup*

'unsaved\_changes': bool, *# Cambios sin guardar (nuevo)*

*# ============================================*

*# PASO -1: DATOS DEL CLIENTE*

*# ============================================*

'client\_data': {

'client\_name': str, *# Nombre del cliente*

'contact\_person': str, *# Persona de contacto*

'contact\_email': str, *# Email de contacto*

'sensor\_sn': str, *# Número de serie del sensor*

'equipment\_model': str, *# Modelo del equipo*

'location': str, *# Ubicación física*

'notes': str, *# Notas adicionales*

'date': str *# Timestamp del proceso*

},

*# ============================================*

*# PASO 1: DIAGNÓSTICO WSTD*

*# ============================================*

'wstd\_data': {

'df': pd.DataFrame, *# DataFrame original con mediciones WSTD*

'grouped': pd.DataFrame, *# DataFrame agrupado por lámpara*

'spectral\_cols': list, *# Lista de columnas espectrales ['#1', '#2', ...]*

'lamps': list *# Lista de nombres de lámparas*

},

*# ============================================*

*# PASO 2: STANDARD KIT (ACTUALIZADO)*

*# ============================================*

'kit\_data': {

'df': pd.DataFrame, *# DataFrame de nueva lámpara (para paso 6)*

'df\_ref\_grouped': pd.DataFrame, *# Espectros referencia (agrupados por ID)*

'df\_new\_grouped': pd.DataFrame, *# Espectros nueva lámpara (agrupados por ID)*

'spectral\_cols': list, *# Columnas espectrales*

'lamp\_ref': str, *# "Referencia" (genérico)*

'lamp\_new': str, *# "Nueva" (genérico)*

'common\_ids': list, *# IDs comunes entre archivos*

'mean\_diff': np.array *# Vector de corrección (se agrega en paso 3)*

},

*# ============================================*

*# PASO 2: SELECCIÓN DE MUESTRAS*

*# ============================================*

'selected\_ids': list, *# IDs confirmados para corrección*

'pending\_selection': list, *# IDs pendientes de confirmar (UI)*

*# ============================================*

*# PASO 4: BASELINE*

*# ============================================*

'baseline\_data': {

'ref\_spectrum': np.array, *# Espectro del baseline (array de floats)*

'header': np.array, *# Cabecera .ref (3 valores) o None*

'df\_baseline': pd.DataFrame,*# DataFrame CSV completo o None*

'origin': str *# 'ref' o 'csv'*

}

}

**📊 Formato de Datos Espectrales**

**DataFrame de Mediciones (TSV)**

python

*# Estructura de df\_ref o df\_new*

df = pd.DataFrame({

'ID': ['Sample01', 'Sample01', 'Sample02', ...],

'Note': ['LampOld', 'LampOld', 'LampOld', ...], *# Ya no se usa en nueva versión*

'#1': [0.123, 0.125, 0.121, ...], *# Canal espectral 1*

'#2': [0.234, 0.236, 0.232, ...], *# Canal espectral 2*

'#3': [0.345, 0.347, 0.343, ...], *# Canal espectral 3*

*# ... hasta #256 (típicamente)*

})

**DataFrame Agrupado**

python

*# df\_ref\_grouped o df\_new\_grouped (índice = ID)*

df\_grouped = pd.DataFrame({

'#1': [0.124, 0.235, ...], *# Promedio de mediciones repetidas*

'#2': [0.235, 0.346, ...],

*# ... resto de canales*

}, index=['Sample01', 'Sample02', ...])

**Vector de Corrección**

python

*# mean\_diff - numpy array*

mean\_diff = np.array([

0.001, *# Corrección canal 1*

0.002, *# Corrección canal 2*

0.003, *# Corrección canal 3*

*# ... hasta canal 256*

])

*# Longitud: len(spectral\_cols)*

**Baseline Spectrum**

python

*# ref\_spectrum - numpy array*

baseline = np.array([

2456.301, *# Intensidad canal 1*

2457.102, *# Intensidad canal 2*

2458.903, *# Intensidad canal 3*

*# ... hasta canal 256*

])

```

---

*## 🔄 Flujo de Variables entre Pasos*

```

PASO -1: client\_data

↓ (guarda)

st.session\_state.client\_data

PASO 1: wstd\_data

↓ (guarda)

st.session\_state.wstd\_data

PASO 2: kit\_data (2 archivos TSV)

↓ (carga TSV ref + TSV new)

↓ (agrupa por ID)

↓ (encuentra comunes)

↓ (usuario selecciona muestras)

↓ (guarda)

st.session\_state.kit\_data

st.session\_state.selected\_ids

PASO 3: corrección

↓ (lee kit\_data + selected\_ids)

↓ (calcula mean\_diff)

↓ (actualiza)

st.session\_state.kit\_data['mean\_diff']

PASO 4: baseline

↓ (carga .ref o .csv)

↓ (guarda)

st.session\_state.baseline\_data

PASO 5: exportación

↓ (lee todo)

↓ (aplica corrección)

↓ (genera archivos + informe)

**🔑 Variables Clave por Módulo**

**config.py**

python

PAGE\_CONFIG: dict *# Configuración Streamlit*

STEPS: list *# Nombres de los pasos*

BASELINE\_PATHS: dict *# Rutas de archivos baseline*

WSTD\_THRESHOLDS: dict *# Umbrales de diagnóstico*

DIAGNOSTIC\_STATUS: dict *# Estados con iconos/colores*

DEFAULT\_CSV\_METADATA: dict *# Metadatos por defecto para CSV*

SPECIAL\_IDS: dict *# IDs especiales ('WSTD')*

PLOT\_CONFIG: dict *# Configuración de gráficos*

INSTRUCTIONS: dict *# Textos de instrucciones*

MESSAGES: dict *# Mensajes de error/éxito*

REPORT\_STYLE: str *# CSS para informes HTML*

**session\_manager.py - Funciones Principales**

python

*# Inicialización*

initialize\_session\_state()

*# Navegación*

go\_to\_step(step\_number: int)

go\_to\_next\_step()

go\_to\_previous\_step()

get\_current\_step() -> int

*# Guardar datos*

save\_client\_data(\*\*kwargs)

save\_wstd\_data(df, grouped, spectral\_cols, lamps)

save\_kit\_data(df, df\_ref\_grouped, df\_new\_grouped, spectral\_cols,

lamp\_ref, lamp\_new, common\_ids)

save\_baseline\_data(ref\_spectrum, header, df\_baseline, origin)

update\_kit\_data\_with\_correction(mean\_diff)

*# Selección de muestras*

update\_selected\_samples(selected\_ids: list)

update\_pending\_selection(pending\_ids: list)

*# Validadores*

has\_client\_data() -> bool

has\_wstd\_data() -> bool

has\_kit\_data() -> bool

has\_baseline\_data() -> bool

has\_correction\_data() -> bool

*# Reset*

reset\_session\_state()

**core/file\_handlers.py - Funciones**

python

*# Lectura*

load\_tsv\_file(file) -> pd.DataFrame

get\_spectral\_columns(df) -> list

load\_ref\_file(file) -> tuple[np.array, np.array] *# (header, spectrum)*

load\_csv\_baseline(file) -> tuple[pd.DataFrame, np.array] *# (df, spectrum)*

*# Escritura*

export\_ref\_file(spectrum, header) -> bytes

export\_csv\_file(spectrum, df\_baseline=None) -> str

**core/spectral\_processing.py - Funciones**

python

*# Agrupación*

group\_measurements\_by\_lamp(df, spectral\_cols, lamp\_ref, lamp\_new)

-> tuple[pd.DataFrame, pd.DataFrame]

*# Búsqueda*

find\_common\_samples(df\_ref\_grouped, df\_new\_grouped) -> pd.Index

*# Cálculos*

calculate\_spectral\_correction(df\_ref\_grouped, df\_new\_grouped, selected\_ids=None)

-> np.array

apply\_baseline\_correction(baseline\_spectrum, correction\_vector)

-> np.array

simulate\_corrected\_spectra(df\_kit, spectral\_cols, lamp\_new,

baseline\_original, baseline\_corrected)

-> pd.DataFrame

**utils/validators.py - Funciones**

python

validate\_wstd\_measurements(df\_wstd, lamps) -> bool

validate\_common\_samples(common\_ids) -> bool

validate\_dimension\_match(baseline\_length, tsv\_length) -> bool

**utils/plotting.py - Funciones**

python

*# Gráficos principales*

plot\_wstd\_spectra(df\_wstd\_grouped, spectral\_cols, lamps) -> go.Figure

plot\_kit\_spectra(df\_ref\_grouped, df\_new\_grouped, spectral\_cols,

lamp\_ref, lamp\_new, common\_ids) -> go.Figure

plot\_correction\_differences(df\_diff, common\_ids, selected\_ids) -> go.Figure

plot\_baseline\_comparison(baseline\_original, baseline\_corrected, spectral\_cols) -> go.Figure

plot\_corrected\_spectra\_comparison(df\_ref\_grouped, df\_corrected, spectral\_cols,

lamp\_ref, lamp\_new, sample\_ids, title) -> go.Figure

*# Auxiliares*

plot\_correction\_summary(mean\_diff) -> go.Figure

plot\_baseline\_spectrum(baseline\_spectrum, lamp\_name) -> go.Figure

**📝 Tipos de Archivos Manejados**

**Entrada**

| **Archivo** | **Formato** | **Paso** | **Descripción** |
| --- | --- | --- | --- |
| **WSTD.tsv** | TSV | 1 | Mediciones White Standard sin baseline |
| **Kit\_Ref.tsv** | TSV | 2 | Mediciones de referencia (archivo histórico) |
| **Kit\_New.tsv** | TSV | 2 | Mediciones con nueva lámpara |
| **baseline.ref** | Binario | 4 | Baseline formato antiguo (≤v531) |
| **baseline.csv** | CSV | 4 | Baseline formato nuevo (≥v557) |

**Salida**

| **Archivo** | **Formato** | **Paso** | **Descripción** |
| --- | --- | --- | --- |
| **baseline\_corregido.ref** | Binario | 5 | Baseline corregido formato antiguo |
| **baseline\_corregido.csv** | CSV | 5 | Baseline corregido formato nuevo |
| **correccion\_tabla.csv** | CSV | 3 | Tabla con detalles de corrección |
| **comparacion\_baseline.csv** | CSV | 5 | Comparación original vs corregido |
| **espectros\_corregidos.tsv** | TSV | 5 | TSV con simulación de corrección |
| **Informe.html** | HTML | 5 | Informe completo del proceso |

**🎨 Constantes Importantes**

**Umbrales de Diagnóstico**

python

WSTD\_THRESHOLDS = {

'good': 0.01, *# < 0.01 → Bien calibrado*

'warning': 0.05, *# 0.01-0.05 → Moderado*

'bad': float('inf') *# > 0.05 → Requiere ajuste*

}

**Columnas Espectrales**

python

*# Formato: '#1', '#2', '#3', ..., '#256'*

*# Detectadas automáticamente con: get\_spectral\_columns(df)*

spectral\_cols = ['#1', '#2', '#3', ..., '#256']

**IDs Especiales**

python

SPECIAL\_IDS = {

'wstd': 'WSTD' *# Identificador del White Standard*

}

**🔧 Cambios Recientes (Nueva Versión)**

**Cambio Principal: Dos Archivos TSV Separados**

**Antes:**

python

*# Un solo archivo con columna 'Note'*

df['Note'] = ['LampOld', 'LampOld', 'LampNew', 'LampNew', ...]

lamp\_ref = st.selectbox("Selecciona lámpara ref", lamp\_options)

lamp\_new = st.selectbox("Selecciona lámpara nueva", lamp\_options)

**Ahora:**

python

*# Dos archivos independientes*

ref\_file = st.file\_uploader("TSV Referencia")

new\_file = st.file\_uploader("TSV Nueva")

*# Nombres genéricos*

lamp\_ref = "Referencia"

lamp\_new = "Nueva"

**Impacto en Variables**

python

*# kit\_data mantiene la misma estructura*

kit\_data = {

'df': df\_new\_kit, *# Solo el de nueva lámpara*

'df\_ref\_grouped': ..., *# Del archivo de referencia*

'df\_new\_grouped': ..., *# Del archivo nuevo*

'lamp\_ref': "Referencia", *# ← Genérico (antes era seleccionado)*

'lamp\_new': "Nueva", *# ← Genérico (antes era seleccionado)*

'common\_ids': ...,

'spectral\_cols': ...,

'mean\_diff': ... *# Se agrega en paso 3*

}

```

*### \*\*¿Qué Cambiar en report\_generator.py?\*\**

❌ \*\*NO necesitas cambiar\*\* - Los nombres `lamp\_ref` y `lamp\_new` se siguen usando igual

✅ Ahora solo dirán "Referencia" y "Nueva" en vez de nombres específicos

---

*## 📊 Diagrama de Dependencias de Datos*

```

client\_data (opcional)

↓

wstd\_data (opcional - diagnóstico)

↓

kit\_data ← CRÍTICO

├─ df\_ref\_grouped

├─ df\_new\_grouped

├─ selected\_ids → mean\_diff ← CLAVE

└─ spectral\_cols

↓

baseline\_data ← CRÍTICO

├─ ref\_spectrum

└─ header (si .ref) o df\_baseline (si .csv)

↓

Aplicar corrección:

baseline\_corrected = baseline - mean\_diff

↓

Exportar archivos + informe

**🎯 Variables Obligatorias vs Opcionales**

**Obligatorias (para que funcione el flujo completo)**

python

✅ kit\_data['df\_ref\_grouped']

✅ kit\_data['df\_new\_grouped']

✅ kit\_data['spectral\_cols']

✅ kit\_data['mean\_diff']

✅ baseline\_data['ref\_spectrum']

✅ selected\_ids

**Opcionales**

python

⭕ client\_data (se puede omitir)

⭕ wstd\_data (diagnóstico opcional)

⭕ baseline\_data['header'] (solo para .ref)

⭕ baseline\_data['df\_baseline'] (solo para .csv)

⭕ backup\_done (solo flag de confirmación)

**🔒 Validaciones de Integridad**

python

*# Antes de paso 3*

assert has\_kit\_data()

assert len(selected\_ids) > 0

*# Antes de paso 4*

assert has\_kit\_data()

assert has\_correction\_data()

*# Antes de paso 5*

assert has\_kit\_data()

assert has\_baseline\_data()

assert has\_correction\_data()

assert len(baseline\_data['ref\_spectrum']) == len(kit\_data['spectral\_cols'])

**📌 Notas Finales**

* **Todos los DataFrames espectrales usan columnas #1, #2, ..., #N**
* **Los índices de df\_ref\_grouped y df\_new\_grouped son los IDs de muestra**
* **mean\_diff siempre tiene la misma longitud que spectral\_cols**
* **baseline\_corrected = baseline\_original - mean\_diff** (resta, no suma)
* **Los nombres "Referencia" y "Nueva" son genéricos en la nueva versión**

**Este documento describe la estructura completa de variables y datos de la aplicación actualizada con soporte para dos archivos TSV separados.** 📚✅