



MANUAL TÉCNICO PRE-PERC

Arquitectura, algoritmos, base de datos, persistencia y operación técnica del sistema

Versión	Fecha	Área
3.0	24 de junio de 2026	Oficina de Costos

Alcance documental

Documenta la arquitectura Laravel, los servicios de consolidación, lectura de Excel, reglas de matching, almacenamiento, seguridad, rutas, modelos de dominio, exportación y consideraciones de mantenimiento.

Control del documento

Título	Manual Técnico PRE-PERC
Sistema	PRE-PERC - Sistema de Consolidación de Costos
Módulo	Carga y Consolidación de Horas Extras
Versión	3.0
Fecha	24 de junio de 2026
Área responsable	Oficina de Costos / Soporte técnico institucional
Audiencia	Desarrollo, soporte técnico, infraestructura y mantenimiento del sistema

Historial de cambios

Versión	Fecha	Descripción	Responsable
3.0	24 de junio de 2026	Actualización documental con identidad PRE-PERC, formato profesional, control operativo y gráficos de apoyo.	Oficina de Costos

Resumen ejecutivo

Este manual técnico explica cómo está construido PRE-PERC y qué componentes intervienen en la consolidación mensual. Incluye arquitectura general, estructura de carpetas, modelos de dominio, seguridad, rutas, almacenamiento, procesamiento Excel, reglas de matching y trazabilidad.

Tabla de contenido

Este índice resume las secciones principales del documento. En Word se puede actualizar o complementar con numeración de páginas si se requiere una tabla automática.

1. Objetivo técnico

2. Resumen funcional

3. Tecnologías utilizadas

- Backend
- Base de datos
- Procesamiento de Excel
- Frontend
- Documentación

4. Por qué estas tecnologías

- Laravel
- PhpSpreadsheet
- Alpine.js
- SQL Server

5. Arquitectura general

6. Estructura de carpetas

- Carpetas relevantes

7. Modelos de dominio

- `ConsolidationSession`
- `ConsolidationFormUpload`
- `ConsolidationLog`
- `ConsolidationUploadMutation`
- `TemplateColumnMapping`

8. Estados y workflow

- Estados de sesión
- Estados de upload
- Orden del workflow

9. Seguridad

- 9.1 Acceso inicial
- 9.2 Candado por sesión
- 9.3 Inactividad
- 9.4 Protección del directorio raíz

10. Rutas HTTP principales

- Acceso
- Sesiones
- Control de sesión
- MOV de planilla
- Plantillas
- Formularios
- Validaciones
- Exportación

11. Estructura de almacenamiento

12. Configuración central

13. Resolución del MOV de planilla

- Lector

PRE-PERC | Manual Técnico

Qué hace

Clasificación

14. Generación mensual de plantillas

Responsabilidades

Regla mensual importante

15. Resolución de la plantilla de horas

Qué detecta

Particularidad

16. Resolución de la plantilla de pagos

Columnas clave

Qué hace

17. Validación del tipo de formulario

Objetivo

Método

18. Lectores de formularios

Lógica general de los lectores

19. Normalización

Responsabilidades

Ejemplos

20. Algoritmo de búsqueda del colaborador

Importante

Orden de matching

Umbrales

Qué evita

21. Resolución de columna destino

Regla principal

Reglas especiales

22. Procesador central

Pipeline por upload

23. Persistencia de cambios

`UploadMutationService`

`WorkbookPersistenceService`

`WorkingTemplateRebuilder`

24. Logs y trazabilidad

Ejemplos de pasos logueados

25. Validación manual

Acciones posibles

26. Dashboard y analítica

Produce

27. Exportación

Tipos

Resultado

28. Técnicas para consistencia y seguridad funcional

29. Problemas típicos y cómo se mitigan

Encabezados confundidos como personas

Cédulas con formatos distintos

Formularios en tipo incorrecto

Personas sin `ORD`

30. Consideraciones de rendimiento

31. Limitaciones conocidas

32. Posibles mejoras futuras

33. Resumen técnico corto

Vista general del sistema

PRE-PERC es una aplicación Laravel sobre SQL Server que procesa archivos Excel reales mediante PhpSpreadsheet. El sistema opera como un pipeline controlado por sesiones mensuales, con acceso por clave, piloto único, validación manual asistida, mutaciones reversibles y almacenamiento de workbooks por sesión.

Nota de denominación técnica

PRE-PERC es la identidad visible del sistema. Algunas rutas internas conservan el nombre legado costos/consolidador-horas por compatibilidad con el despliegue existente.

Arquitectura general PRE-PERC

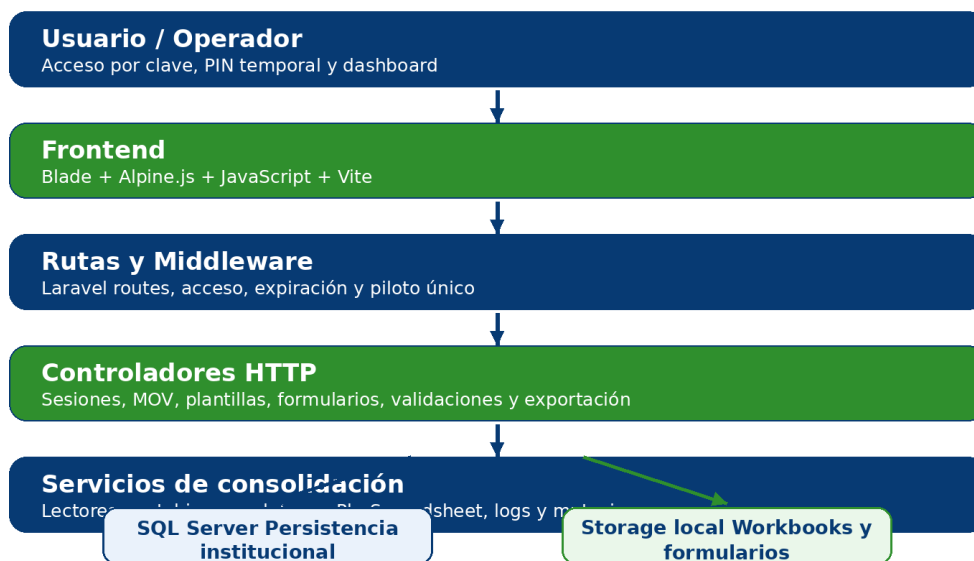


Figura 1. Arquitectura general del sistema PRE-PERC.

Estrategia determinística de búsqueda del colaborador



Figura 2. Estrategia escalonada para búsqueda de colaboradores.

Modelo funcional de datos y trazabilidad

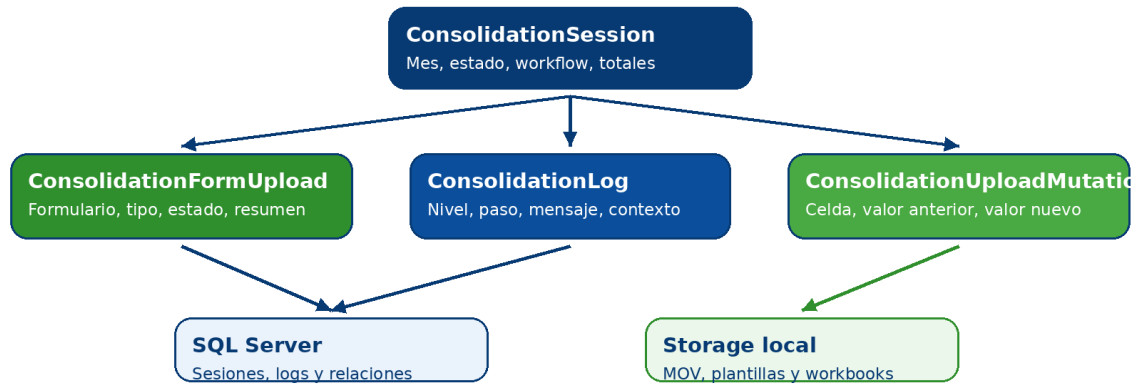


Figura 3. Modelo funcional de datos, logs, mutaciones y almacenamiento.

1. Objetivo técnico

El sistema automatiza el cierre mensual de horas extras para la Oficina de Costos del Hospital del Niño José Renán Esquivel.

Su objetivo es transformar múltiples archivos Excel heterogéneos en dos salidas consistentes:

- una **plantilla de horas**
- una **plantilla de pagos**

La plataforma agrega:

- control de acceso por clave
- piloto unico por sesión
- generación mensual basada en MOV de planilla
- validación manual asistida
- trazabilidad por logs y mutaciones
- exportación final de horas y pagos

2. Resumen funcional

El sistema opera por sesiones mensuales y sigue este pipeline:

1. crear o retomar sesión
2. cargar MOV de planilla
3. clasificar ingresos, egresos, bonificaciones y actualizaciones
4. generar plantillas del mes
5. validar y corregir plantillas
6. cargar formularios por tipo
7. procesar horas y pagos
8. revisar dashboard final y exportar

3. Tecnologías utilizadas

Backend

- **PHP 8.2**

Se usa porque el sistema corre sobre XAMPP en entorno institucional y Laravel 12 requiere PHP moderno.

- **Laravel 12**

Se usa por su contenedor de servicios, enrutamiento, middleware, validación, Eloquent y facilidad para separar lógica operativa en servicios.

Base de datos

- **SQL Server**
- driver **sqlsrv**
- **ODBC Driver 18**

Se usa porque el entorno del hospital ya trabaja con esa base institucional.

Procesamiento de Excel

- **PhpSpreadsheet**

Se usa porque el problema central del sistema es leer, transformar, recalcular y guardar libros Excel reales con hojas, fórmulas, estilos y celdas concretas.

Frontend

- **Blade**
- **JavaScript vanilla**
- **Alpine.js**
- **CSS**
- **Vite**

Se usa esta combinación porque el sistema necesita interacción rica sin construir una SPA pesada.

Documentación

- **Python**
- **ReportLab**

Se usan para generar los manuales PDF desde archivos Markdown editables.

4. Por qué estas tecnologías

Laravel

Permite:

- separar controladores y servicios
- proteger rutas con middleware
- trabajar con SQL Server sin escribir todo a mano
- mantener el flujo por sesiones y modelos de manera clara

PhpSpreadsheet

Permite:

- detectar hojas y encabezados
- leer valores, fórmulas y formatos
- insertar columnas
- escribir en coordenadas exactas
- guardar copias de trabajo sin perder la estructura del archivo

Alpine.js

Permite:

- controlar el dashboard por pasos
- reaccionar a validaciones, filtros y progreso
- evitar un framework frontend mayor para una app interna

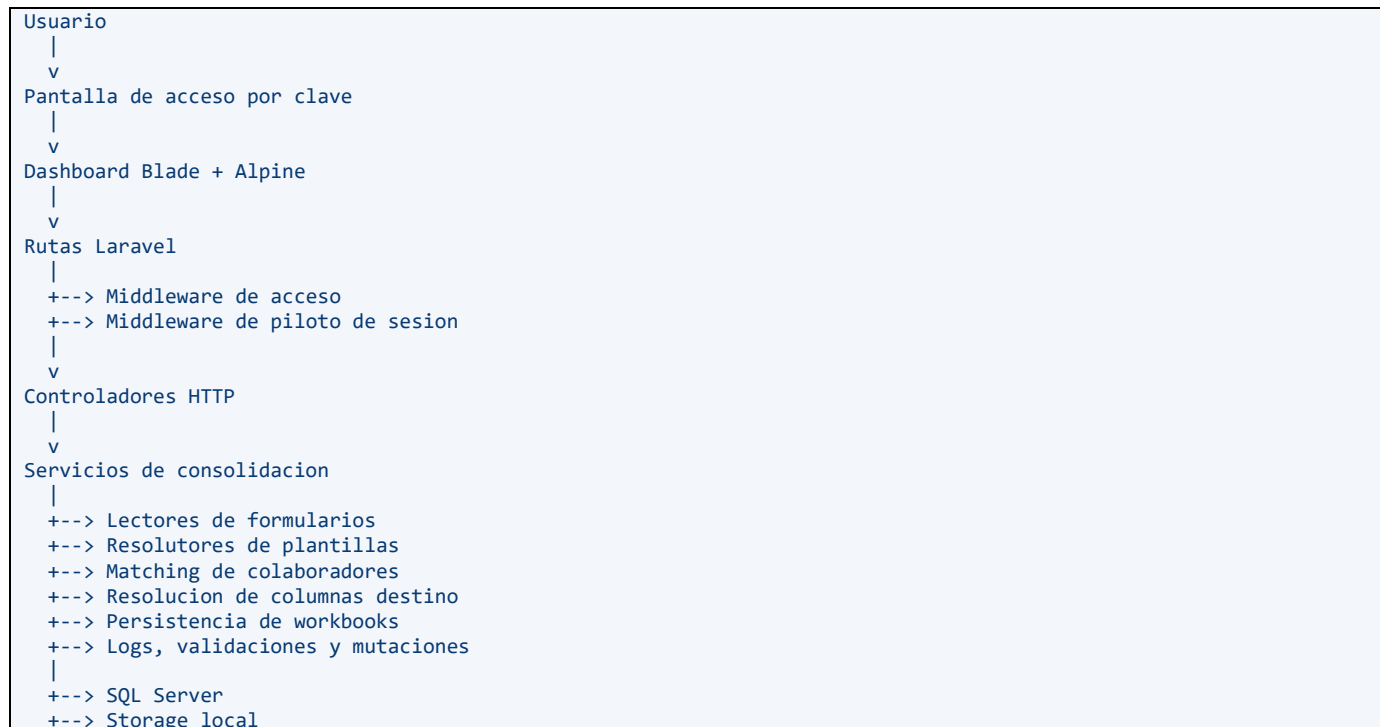
SQL Server

Permite:

- persistencia operativa centralizada
- integridad por relaciones

- trazabilidad de sesiones, formularios, logs y mutaciones

5. Arquitectura general



6. Estructura de carpetas

El proyecto principal esta en:

C:\xampp\htdocs\costos\consolidador-horas

Carpetas relevantes

- `app/`

Código PHP principal.

- `app/Http/Controllers/`

Puntos de entrada HTTP del sistema.

- `app/Http/Middleware/`

Reglas de acceso, expiración y propiedad de sesión.

- `app/Models/`

Modelos Eloquent de sesiones, formularios, logs y mutaciones.

- `app/Services/Consolidation/`

Motor del negocio: generación del MOV, resolución de plantillas, procesamiento, matching, validación y persistencia.

- `app/Services/Consolidation/FormReaders/`

Lectores especializados por tipo de formulario Excel.

- [app/Services/Consolidation/Support/](#)

Utilidades transversales como normalizacion y logging.

- [bootstrap/](#)

Arranque de la aplicación y registro de middleware.

- [config/](#)

Configuración del sistema, incluyendo tipos de formulario y parámetros de acceso.

- [database/](#)

Migraciones, factories y seeders.

- [docs/](#)

Manuales y generador PDF.

- [public/](#)

Punto de entrada web, assets y reglas de acceso.

- [resources/views/](#)

Vistas Blade del dashboard y acceso.

- [routes/](#)

Definición de rutas web.

- [storage/](#)

Archivos temporales, plantillas generadas, sesiones y locks.

- [tests/](#)

Pruebas del sistema.

- [vendor/](#)

Dependencias de Composer.

7. Modelos de dominio

`ConsolidationSession`

Representa el mes de trabajo.

Contiene:

- periodo
- estado
- plantillas generadas
- snapshot estructural
- totales
- workflow por pasos

`ConsolidationFormUpload`

Representa un formulario subido.

Contiene:

- tipo
- nombre original
- estado

- resumen de procesamiento

`ConsolidationLog`

Guarda:

- nivel
- paso
- mensaje
- contexto

`ConsolidationUploadMutation`

Guarda cada mutacion de celdas para:

- revertir
- reconstruir
- auditar

`TemplateColumnMapping`

Mantiene compatibilidad con mapeos dónde todavía hacen falta reglas por tipo o origen, aunque la mayoría del sistema actual usa resolución automática.

8. Estados y workflow

Estados de sesión

- draft
- ready
- processing
- processed
- exported

Estados de upload

- pending
- processing
- processed
- failed
- reverted

Orden del workflow

En C:\xampp\htdocs\costos\consolidador-horas\app\Models\ConsolidationSession.php se definen:

9. session
10. movement
11. preview
12. downloads
13. adjustments
14. forms
15. process

Visualmente existe también el dashboard final como cierre ejecutivo.

9. Seguridad

9.1 Acceso inicial

La ruta principal esta protegida por `C:\xampp\htdocs\costos\consolidador-horas\app\Http\Middleware\EnsureDashboardAccess.php`.

Controla:

- clave de acceso
- expiración de acceso
- redirección a `/access`
- respuesta `401` para peticiones AJAX vencidas

9.2 Candado por sesión

`SessionPilotLockService` permite un solo piloto activo por sesión.

Datos relevantes:

- PIN temporal
- `client_id`
- `pilot_name`
- `locked_at`
- `last_seen_at`

9.3 Inactividad

El frontend muestra advertencia antes del cierre por inactividad y libera la sesión si el operador no confirma continuidad.

9.4 Proteccion del directorio raíz

El directorio `C:\xampp\htdocs\costos` tiene `.htaccess` para:

- evitar listado de directorios
- redirigir a `consolidador-horas`

10. Rutas HTTP principales

Acceso

- `GET /access`
- `POST /access`
- `POST /access/logout`

Sesiones

- `POST /sessions`
- `GET /sessions/{session}`
- `PUT /sessions/{session}`

PRE-PERC | Manual Técnico

- POST /sessions/{session}/update
- POST /sessions/{session}/reset
- POST /sessions/{session}/delete

Control de sesión

- POST /sessions/{session}/lock/claim
- POST /sessions/{session}/lock/heartbeat
- POST /sessions/{session}/lock/release
- POST /sessions/{session}/workflow-stage

MOV de planilla

- POST /sessions/{session}/payroll-movement
- POST /sessions/{session}/payroll-movement/review

Plantillas

- POST /sessions/{session}/master-template
- POST /sessions/{session}/payment-template
- GET /sessions/{session}/generated-template

Formularios

- POST /sessions/{session}/forms
- POST /sessions/{session}/forms/{upload}/delete
- POST /sessions/{session}/forms/{upload}/revert
- POST /sessions/{session}/process

Validaciones

- GET /sessions/{session}/validations/{log}/candidates
- POST /sessions/{session}/validations/{log}/assign
- POST /sessions/{session}/validations/{log}/ignore

Exportación

- GET /sessions/{session}/export

11. Estructura de almacenamiento

Cada sesión usa una carpeta como:

```
storage/app/consolidations/{uuid}/
```

Subcarpetas habituales:

```
movement/payroll-movement.xlsx  
template/master-template.xlsx  
payment-template/master-template.xlsx  
working/consolidated-working.xlsx  
working/consolidated-payments.xlsx  
forms/{archivos_subidos}
```

12. Configuración central

La configuración principal vive en:

C:\xampp\htdocs\costos\consolidador-horas\config\consolidation.php

Define:

- tipos de formulario
- lector asignado a cada tipo
- reglas de validación por nombre de archivo y hoja
- fallbacks de columna destino
- plantillas base del MOV
- directorios de trabajo
- parámetros de acceso y lock

13. Resolución del MOV de planilla

Lector

El archivo se interpreta en C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\PayrollMovementReader.php.

Qué hace

- detecta la fila real de encabezados
- valida que el archivo tenga estructura de MOV
- evita confundir la plantilla de pagos con el MOV
- toma cédula de columnas E, F, G
- toma salario base desde J
- toma adicional desde K
- toma observación desde L
- clasifica cada fila

Clasificación

Las acciones base son:

- add
- remove
- update
- ignore

La clasificación usa palabras clave configuradas y reglas de observación.

14. Generación mensual de plantillas

El servicio principal es:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\PayrollMovementTemplateService.php

Responsabilidades

- guardar el MOV
- buscar las plantillas base del mes anterior
- si no existen, caer a las plantillas oficiales de referencia
- abrir plantilla de horas y pagos
- aplicar ingresos, egresos, bonificaciones y actualizaciones
- guardar plantillas nuevas
- crear copias de trabajo
- construir el snapshot del MOV
- avanzar el workflow al paso de validación

Regla mensual importante

Cada mes nuevo usa como base el mes anterior ya generado, no siempre enero. Esto permite encadenar las plantillas mes a mes.

15. Resolución de la plantilla de horas

El análisis estructural vive en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\TemplateStructureResolver.php

Qué detecta

- hoja valida
- fila de encabezados
- fila inicial de datos
- columna EMPLEADO
- columnas ORD
- columnas EXT
- índice de colaboradores

Particularidad

La celda EMPLEADO puede venir como:

- CEDULA_NOMBRE
- POSICION_CEDULA_NOMBRE
- POSICION_CEDULA_NOMBRE APELLIDO

El resolver usa `TextNormalizer` para separar posicion, cédula y nombre.

16. Resolución de la plantilla de pagos

El análisis estructural vive en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\PaymentTemplateStructureResolver.php

Columnas clave

- `Identificacion`

PRE-PERC | Manual Técnico

- Nombre
- Salario Base
- Bonificaciones
- HORAS EXTRAS
- Sal+Boni+Extras

Qué hace

- detecta o crea columnas faltantes
- limpia Bonificaciones y HORAS EXTRAS cuando se genera el mes
- aplica fórmulas de total
- construye un índice de colaboradores para pagos

17. Validación del tipo de formulario

La validación previa se hace en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\FormUploadValidationService.php

Objetivo

Evitar que un operador suba:

- un formulario en el tipo equivocado
- la plantilla de pagos como si fuera MOV
- un archivo cuya estructura no coincide con el tipo seleccionado

Método

- intenta parsear el archivo con el lector esperado
- usa pistas por nombre de archivo y hojas
- compara coincidencias posibles
- si no hay seguridad, rechaza la carga con mensaje claro

18. Lectores de formularios

Los lectores se resuelven desde:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\FormReaders\FormReaderRegistry.php

Tipos actuales:

- MedicalResidentsFormReader
- AdministrativeStaffFormReader
- HolidayShiftFormReader
- MedicalSpecialistsFormReader
- MedicalSpecialistsUrgenciesFormReader
- ContractPayrollFormReader
- NursesWorkbookFormReader

Lógica general de los lectores

Cada lector:

- detecta encabezados y hoja valida
- descarta subtotales, firmas y encabezados falsos
- extrae nombre, cédula, área, horas, turnos y monto
- devuelve filas estandarizadas para el procesador

19. Normalizacion

La normalizacion vive en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\Support\TextNormalizer.php

Responsabilidades

- normalizar cédulas
- generar variantes de cédula
- normalizar nombres
- detectar áreas
- validar si una cadena parece persona o encabezado
- convertir turnos y expresiones mixtas a horas

Ejemplos

- 4-770-1862 y 407701862 se reconocen como equivalentes
- 3T+4H se convierte en 28
- 1.5H se interpreta como 1 hora 30 minutos

20. Algoritmo de búsqueda del colaborador

La lógica principal esta en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\EmployeeMatchingService.php

Importante

El sistema **no usa una búsqueda recursiva**. Usa una estrategia escalonada y determinística.

Orden de matching

16. cédula exacta
17. nombre exacto
18. nombre compatible por tokens
19. cédula aproximada por variantes
20. nombre similar con umbral

Umbrales

- similitud minima: 0.82
- diferencia minima entre primer y segundo candidato: 0.08

Qué evita

- asignaciones por parecido debil
- borrar o actualizar a una persona incorrecta
- confundir encabezados con colaboradores

21. Resolución de columna destino

La lógica vive en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\TargetColumnResolver.php

Regla principal

- localizar la ORD activa del colaborador
- escribir en la EXT contigua

Reglas especiales

Si no existe ninguna ORD activa, se aplican fallbacks según tipo:

- Medicos Especialistas: columna CO
- Medicos Especialistas (Urgencias): columna CO
- Planilla Contrato: columna BP

22. Procesador central

El motor esta en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\ConsolidationProcessor.php

Pipeline por upload

21. marca el upload como `processing`
22. limpia mutaciones previas
23. usa el lector correcto
24. recorre filas útiles
25. hace matching de colaborador
26. resuelve destino en horas
27. aplica monto en pagos
28. genera logs
29. guarda mutaciones
30. persiste workbooks
31. actualiza progreso

23. Persistencia de cambios

`UploadMutationService`

Guarda celdas tocadas:

- coordenada

- valor anterior
- tipo anterior
- valor nuevo
- delta

`WorkbookPersistenceService`

Guarda de forma segura ambos libros:

- horas
- pagos

`WorkingTemplateRebuilder`

Permite reconstruir la plantilla de trabajo cuando hace falta rehacer consistencia.

24. Logs y trazabilidad

El sistema usa:

- SessionLogService
- tabla consolidation_logs

Ejemplos de pasos logueados

- payroll-movement
- process-upload
- match-employee
- apply-hours
- apply-payment
- resolve-mapping
- export

Esto permite:

- auditoria
- soporte
- revision manual
- panel de validaciones

25. Validación manual

La resolución de validaciones vive en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\ValidationReviewService.php

y en:

C:\xampp\htdocs\costos\consolidador-horas\app\Http\Controllers\ValidationReviewController.php

Acciones posibles

- buscar coincidencias
- asignar
- sumar

- reemplazar
- omitir

26. Dashboard y analítica

La información enriquecida del dashboard se construye en:

C:\xampp\htdocs\costos\consolidador-horas\app\Services\Consolidation\SessionDetailBuilder.php

Produce

- KPIs
- estado de workflow
- sesiones recientes
- historial del MOV
- validaciones
- resumen por formulario
- analítica por tipo, área, archivo y colaborador

27. Exportación

La salida final se gestiona en:

C:\xampp\htdocs\costos\consolidador-horas\app\Http\Controllers\ExportConsolidationController.php

Tipos

- `kind=hours`
- `kind=payments`

Resultado

- descarga del archivo consolidado
- actualización del estado de sesión
- registro del evento en logs

28. Técnicas para consistencia y seguridad funcional

El sistema se protege con varias capas:

- validación del tipo de archivo
- normalización de nombres y cédulas
- umbrales de similitud
- bloqueo por piloto activo
- control de workflow por pasos
- validación manual cuando no hay certeza
- mutaciones reversibles
- reconstrucción de plantilla
- herencia mensual controlada desde el mes anterior

29. Problemas típicos y cómo se mitigan

Encabezados confundidos como personas

Se mitiga en lectores y `TextNormalizer`.

Cédulas con formatos distintos

Se mitiga con variantes de cédula.

Formularios en tipo incorrecto

Se mitiga con `FormUploadValidationService`.

Personas sin `ORD`

Se mitiga con fallbacks de columnas fijas por tipo.

Cambios manuales posteriores

Se mitiga con el paso de ajuste manual y el rebloqueo del flujo.

30. Consideraciones de rendimiento

El sistema ya incorpora varias medidas:

- `set_time_limit(0)` en procesos pesados
- carga por fases y progreso
- no recalcular todo el dashboard a cada paso
- tratamiento especial de archivos grandes
- persistencia segura de workbooks

31. Limitaciones conocidas

- sigue dependiendo de la calidad estructural del Excel fuente
- si el MOV o un formulario viene muy deformado, puede requerir validación manual
- el export final se realiza desde HTTP y depende del estado de sesión
- el sistema no usa colas asíncronas; el proceso corre dentro del request

32. Posibles mejoras futuras

- mover procesamiento pesado a jobs
- colas reales para MOV y consolidación
- versionado formal de plantillas mensuales
- bitacora diferencial por sesión y por usuario
- pruebas automatizadas más amplias por cada lector
- exportes filtrados del dashboard final

33. Resumen técnico corto

El sistema se basa en:

- Laravel para el flujo y la seguridad
- SQL Server para persistencia institucional
- PhpSpreadsheet para manipular Excel real
- reglas de negocio por tipo de formulario
- matching escalonado por cédula y nombre
- resolución automática de columnas [ORD/EXT](#)
- validación manual cuando no hay seguridad suficiente

No es un motor recursivo. Es un pipeline de consolidación orientado a reglas, normalización y trazabilidad.

Cierre documental

Este manual debe mantenerse alineado con los cambios del sistema PRE-PERC, especialmente cuando se ajusten plantillas, reglas de validación, tipos de formularios, parámetros de acceso, estructura de almacenamiento o criterios de exportación.