**✅ Pre-flight (antes de tocar nada)**

* **Crear rama** de trabajo: git checkout -b refactor/orden-html
* **Backup DB**: copia asistencia.db (o tu origen real).
* **Congelar dependencias**: pip freeze > requirements.txt
* **Activar autoformato** en VS Code (Prettier para HTML/CSS/JS; Black/autopep8 para Python).
* **Encender el reloader** de Flask y probar que todo parte limpio (smoke test básico).

**Fase 1 — Estabilización (lo imprescindible que ya vimos)**

* Quitar cualquier uso de | map("isoformat") en Jinja.
* Asegurar en app.py: dias\_rango\_iso = [d.strftime("%d-%m-%Y") for d in dias\_rango].
* En el template, usar: const diasRango = {{ dias\_rango\_iso | tojson | safe }};
* En JS, **parsear DD-MM-YYYY**: const [dd,mm,yyyy]=d.split('-'); new Date(${yyyy}-${mm}-${dd})
* Corregir **descripciones** de códigos: PF = Permiso por Fallecimiento, PSN = Permiso Post-Natal.
* Verificar el cálculo de **ranking de ausencias**:
  + “Histórico” = entre fecha\_desde y fecha\_hasta.
  + “Mensual” = sólo mes actual del selector.
  + Excluir colaborador **desde el mes siguiente** a su fecha\_egreso.
* Verificar que el **último día** del período carga resumen\_ultimo\_dia correcto.

**Fase 2 — Plantilla base (sin mover CSS/JS aún)**

Objetivo: no repetir <head>, fuentes y contenedor.

* Crear templates/base.html con bloques {% block extra\_css %}, {% block content %}, {% block extra\_js %}.
* Elegir **1** vista **simple** (p.ej. reportes.html) y:
  + Reemplazar su armazón por:
  + {% extends "base.html" %}
  + {% block title %}Reportes{% endblock %}
  + {% block extra\_css %}<style>/\* estilos propios \*/</style>{% endblock %}
  + {% block content %}<!-- body original -->>{% endblock %}
  + {% block extra\_js %}<script>/\* js propio \*/</script>{% endblock %}
  + Probar. Si todo ok, repetir en **una** vista más.
* **No** mover todavía CSS/JS externos (evita cambios masivos).

**Fase 3 — Extraer CSS (por partes)**

* Crear static/css/app.css con **variables de tema** y estilos globales (contenedor, .panel, .btn).
* Copiar SOLO lo global (que se repite) desde las vistas y **no** lo específico.
* En base.html: <link rel="stylesheet" href="{{ url\_for('static', filename='css/app.css') }}">
* Verificar que no cambió el layout.
* (Opcional) Nombres de clases **namespaced** si hay colisiones (.jr-...).

**Fase 4 — Extraer JS (por partes)**

* Crear static/js/app.js para utilidades globales (toasts, helpers).
* Asegurar que tus scripts de página corren al final (DOMContentLoaded o al final del body).
* Si mueves JS de una página:
  + Mantener la **misma** firma de variables Jinja.
  + Incluirlo desde la página:  
    {% block extra\_js %}<script src="{{ url\_for('static', filename='js/reporte\_asistencia.js') }}"></script>{% endblock %}

**Fase 5 — Partials reutilizables**

* Si tienes topbar/back-link repetido: crear templates/partials/\_topbar.html y usar {% include 'partials/\_topbar.html' %}.
* (Opcional) Partials para: **filtros de fecha**, **leyenda estados**, **ranking faltas**.

**Fase 6 — Responsive & UX (seguro)**

* Mantener **tabla completa** en desktop.
* En móvil: **toggle Tabla/Lista** (ya implementado). Confirmar:
  + Navegación por **semana** (7 días).
  + Scroll lateral con **sombras** indicativas.
* Accesibilidad básica:
  + Contraste OK (tema oscuro).
  + prefers-reduced-motion: desactivar animaciones si el usuario lo pide.

**Fase 7 — Performance (cuando ya esté ordenado)**

* Índices DB:
  + asistencia(fecha), asistencia(empleado\_id, fecha), asistencia(codigo\_asistencia\_id)
  + empleados(fecha\_ingreso), empleados(fecha\_egreso)
* Revisión de consultas:
  + Evitar N+1; preferir JOIN + GROUP BY.
  + Limitar SELECT \*; pedir sólo columnas usadas.
* Cache ligera en memoria (p.ej. resumen\_ultimo\_dia del día actual).

**Fase 8 — QA de funcionalidad (checklist por pantalla)**

* **Reporte de Asistencia**
  + Cambiar rango → tabla se actualiza.
  + Histórico/Mensual cambia ranking.
  + Empleado con fecha\_egreso no aparece desde **el mes siguiente**.
  + Códigos y colores correctos (PF/PSN).
  + Vista móvil (lista) muestra semana correcta y navega.
* **Registrar asistencia**
  + Grabar cambios y recargar; datos persisten.
  + Navegación de mes no rompe filtros.
* **Gestionar empleados**
  + Búsqueda y paginación siguen OK.
  + Alta/Baja/Edición sin efectos colaterales.
* **Gestión de turnos**
  + Guardar / copiar mes anterior / aplicar plantilla.

**Fase 9 — Seguridad & robustez**

* Validar request.args: fechas válidas (try/except) y swaps desde>hasta.
* Placeholders SQL (evitar concatenar).
* Manejo de None/vacíos (p.ej. turno\_nombre or 'N/A').
* No exponer rutas internas en el front.

**Fase 10 — Cierre y rollback**

* git status limpio; commits atómicos por fase.
* Merge a main con **squash** o PR revisado.
* Si algo falla, git checkout main y **rollback** con el backup de la DB.

**Tips para ejecutarlo sin dolor**

* Avanza **1 vista por vez**.
* Cambios pequeños + prueba. Si se rompe, **reviertes** sólo ese archivo.
* Mantén **ramas cortas** y PRs pequeños.