Registro Diario de Avances – Sprint 2 Aplicación Web para la gestión de itinerarios

**Fecha:**

**Autor:**

**Versión:** 1.0

# 1. Introducción

Este documento detalla el seguimiento y registro diario del Sprint 1 del proyecto Aplicación Web para la gestión de itinerarios, enfocado en la configuración inicial y desarrollo de la base de datos relacional para la gestión de turistas, usuarios e itinerarios. Se describen las tareas asignadas, su estado, y las estimaciones de esfuerzo realizadas por el equipo de desarrollo, con el objetivo de asegurar la correcta planificación y ejecución del sprint, garantizando la integridad y funcionalidad del sistema desde sus cimientos.

# 2. Lineamientos de Diseño

Para el desarrollo del Sprint 2, se establecieron los siguientes lineamientos de diseño:

* Definir y aplicar una política de respaldos automáticos (frecuencia y retención).
* Configurar y probar la restauración de respaldos en MySQL.
* Ejecutar una prueba piloto con datos de prueba para validar la integridad de la base de datos.
* Verificar rendimiento y consistencia mediante consultas de validación.
* Documentar los procesos de respaldo, restauración y validación para su uso en producción.

# 3. Historias de Usuario y Tareas

| Backlog ID | Historia de Usuario | Tiempo Estimado | Fecha de Inicio | Fecha de Finalización |
| --- | --- | --- | --- | --- |
| HU-19.1 | Configuración de respaldos automáticos | 18 hrs | 24/09/2025 | 26/09/2025 |
| HU-24.1 | Ejecución de prueba piloto | 20 hrs | 26/04/2025 | 30/09/2025 |

# 4. Evidencias de Código

## 4.1 Historia de Usuario 19.1 — Configuración de respaldos automáticos

Archivo: TravelDesk\backend\scripts\backup\backup-full.ps1

Archivo: TravelDesk\backend\scripts\backup\backup-full.ps1

$ErrorActionPreference = "Stop"

. "$PSScriptRoot\\backup.config.ps1"

# Respaldo completo simple a .sql con retención básica

$fecha = Get-Date -Format "yyyyMMdd\_HHmmss"

$dest = Join-Path (Join-Path $BACKUP\_DIR "full") ("$($DB\_NAME)\_$fecha.sql")

$mysqldump = Join-Path $MYSQL\_BIN "mysqldump.exe"

if (!(Test-Path $mysqldump)) { throw "No se encontró mysqldump.exe en '$mysqldump'" }

$dumpArgs = @(

"--host=$DB\_HOST",

"--port=$DB\_PORT",

"--user=$DB\_USER",

"--password=$DB\_PASSWORD",

"--databases", $DB\_NAME,

"--single-transaction"

)

& $mysqldump @dumpArgs | Set-Content -Path $dest -Encoding UTF8

# Retener solo los últimos N días

Get-ChildItem (Join-Path $BACKUP\_DIR "full") -Filter "\*.sql" |

Where-Object { $\_.LastWriteTime -lt (Get-Date).AddDays(-$RETENCION\_DIAS\_COMPLETOS) } |

Remove-Item -Force

Write-Host "Respaldo completo listo: $dest"

Archivo: TravelDesk\backend\scripts\backup\backup-incremental.ps1

$ErrorActionPreference = "Stop"

. "$PSScriptRoot\\backup.config.ps1"

# Incremental simple: copia binlogs nuevos si existe $BINLOG\_BASENAME

if ([string]::IsNullOrWhiteSpace($BINLOG\_BASENAME)) {

throw "Configura BINLOG\_BASENAME y habilita log-bin en MySQL para usar incrementales."

}

$origen = Get-ChildItem -Path $MYSQL\_DATA\_DIR -Filter ("$BINLOG\_BASENAME.\*") | Sort-Object Name

if (-not $origen) { throw "No se encontraron binlogs en '$MYSQL\_DATA\_DIR'" }

$destDir = Join-Path $BACKUP\_DIR "incremental"

foreach ($f in $origen) {

$destFile = Join-Path $destDir $f.Name

if (!(Test-Path $destFile)) {

Copy-Item $f.FullName $destFile -Force

}

}

# Retener solo los últimos N días por fecha de modificación

Get-ChildItem $destDir -File |

Where-Object { $\_.LastWriteTime -lt (Get-Date).AddDays(-$RETENCION\_DIAS\_INCREMENTALES) } |

Remove-Item -Force

Write-Host "Incremental listo en: $destDir"

Archivo: TravelDesk\backend\scripts\backup\backup-verify-restore.ps1

$ErrorActionPreference = "Stop"

. "$PSScriptRoot\\backup.config.ps1"

# Verificación simple de restauración del último respaldo completo (.sql)

$mysql = Join-Path $MYSQL\_BIN "mysql.exe"

if (!(Test-Path $mysql)) { throw "No se encontró mysql.exe en '$mysql'" }

$fullDir = Join-Path $BACKUP\_DIR "full"

$ultimo = Get-ChildItem $fullDir -Filter "\*.sql" | Sort-Object LastWriteTime -Descending | Select-Object -First 1

if (-not $ultimo) { throw "No se encontró ningún archivo .sql en '$fullDir'" }

$dbPrueba = "${DB\_NAME}\_prueba\_restore"

& $mysql -h $DB\_HOST -P $DB\_PORT -u $DB\_USER -p$DB\_PASSWORD -e "DROP DATABASE IF EXISTS ``$dbPrueba``; CREATE DATABASE ``$dbPrueba`` CHARACTER SET utf8mb4;" | Out-Null

# Importar el dump

# PowerShell no implementa el operador de redirección de entrada '<'. Usamos cmd.exe para hacer la redirección de forma segura.

$cmd = '"{0}" -h {1} -P {2} -u {3} -p{4} {5} < "{6}"' -f $mysql, $DB\_HOST, $DB\_PORT, $DB\_USER, $DB\_PASSWORD, $dbPrueba, $ultimo.FullName

cmd.exe /c $cmd

Write-Host "Restauración de prueba completada en: $dbPrueba (desde $($ultimo.Name))"

## 4.2 Historia de Usuario 24.1 — Ejecución de prueba piloto

Archivo: datosprueba.sql

-- Usuarios

INSERT INTO usuarios (nombre\_usuario, email, password, rol)

VALUES ('admin', 'admin@gmail.com', '123456', 'admin'),

('agente', 'agente@gmail.com', '123456', 'agente'),

('cliente', 'cliente@gmail.com', '123456', 'cliente');

-- Grupos

INSERT INTO grupos (nombre, descripcion)

VALUES ('Grupo Cusco Setiembre', 'Turistas que visitan Cusco en setiembre'),

('Grupo Machu Octubre', 'Grupo exclusivo para Machu Picchu en octubre');

-- Estados presupuesto

INSERT INTO estados\_presupuesto (nombre\_estado)

VALUES ('Pendiente'), ('Aprobado'), ('Rechazado');

-- Itinerarios

INSERT INTO itinerarios (id\_grupo, fecha\_inicio, fecha\_fin, estado\_presupuesto\_id)

VALUES (1, '2025-09-25', '2025-09-28', 1),

(2, '2025-10-05', '2025-10-07', 2);

-- Programas

INSERT INTO programas (nombre, descripcion, tipo, duracion, costo)

VALUES ('City Tour Cusco', 'Recorrido por la ciudad de Cusco', 'tour', 1, 150.00),

('Visita Machu Picchu', 'Tour completo al santuario', 'machupicchu', 1, 300.00),

('Trekking Montaña de 7 Colores', 'Caminata guiada', 'actividad', 1, 200.00);

-- Itinerario Programas

INSERT INTO itinerario\_programas (id\_itinerario, id\_programa, fecha, hora\_inicio, hora\_fin)

VALUES (1, 1, '2025-09-25', '09:00:00', '13:00:00'),

(1, 3, '2025-09-26', '06:00:00', '15:00:00'),

(2, 2, '2025-10-06', '05:30:00', '17:00:00');

-- Turistas

INSERT INTO turistas (id\_usuario, nombre, apellido, dni, nacionalidad, fecha\_nacimiento, genero)

VALUES (3, 'Juan', 'Perez', '12345678', 'Peruano', '1990-01-01', 'M'),

(3, 'Maria', 'Lopez', '87654321', 'Argentina', '1992-03-15', 'F');

-- Itinerario - Turistas

INSERT INTO itinerario\_turistas (id\_itinerario, id\_turista)

VALUES (1, 1), (1, 2), (2, 2);

-- Transportes

INSERT INTO transportes (empresa, tipo, capacidad, contacto)

VALUES ('Peru Rail', 'tren', 100, '999-888-777'),

('InkaBus', 'bus', 50, '988-777-666');

-- Detalle Transporte Itinerario

INSERT INTO detalle\_transporte\_itinerario (id\_itinerario\_programa, id\_transporte, horario\_recojo, lugar\_recojo)

VALUES (1, 2, '08:00:00', 'Plaza de Armas'),

(3, 1, '04:30:00', 'Estación de tren');

-- Detalle Machu Picchu Itinerario

INSERT INTO detalle\_machu\_itinerario (id\_itinerario\_programa, empresa\_tren, horario\_tren\_ida, horario\_tren\_retor, nombre\_guia, ruta, tiempo\_visita)

VALUES (3, 'Peru Rail', '06:00:00', '18:00:00', 'Carlos Huamán', 'Aguas', '4 horas');

Archivo: consultasvalidacion.sql

-- Itinerarios con sus grupos y estado

SELECT i.id\_itinerario, g.nombre AS grupo, e.nombre\_estado, i.fecha\_inicio, i.fecha\_fin

FROM itinerarios i

JOIN grupos g ON i.id\_grupo = g.id\_grupo

JOIN estados\_presupuesto e ON i.estado\_presupuesto\_id = e.id\_estado;

-- Programas por itinerario

SELECT ip.id\_itinerario\_programa, i.id\_itinerario, p.nombre, ip.fecha, ip.hora\_inicio, ip.hora\_fin

FROM itinerario\_programas ip

JOIN programas p ON ip.id\_programa = p.id\_programa

JOIN itinerarios i ON ip.id\_itinerario = i.id\_itinerario;

-- Turistas en cada itinerario

SELECT it.id\_itinerario, t.nombre, t.apellido, t.nacionalidad

FROM itinerario\_turistas it

JOIN turistas t ON it.id\_turista = t.id\_turista;

-- Transporte asignado

SELECT dti.id\_detalle\_transporte, ip.id\_itinerario\_programa, tr.empresa, tr.tipo, dti.horario\_recojo, dti.lugar\_recojo

FROM detalle\_transporte\_itinerario dti

JOIN transportes tr ON dti.id\_transporte = tr.id\_transporte

JOIN itinerario\_programas ip ON dti.id\_itinerario\_programa = ip.id\_itinerario\_programa;

-- Detalle Machu Picchu

SELECT dmi.id\_itinerario\_programa, dmi.empresa\_tren, dmi.horario\_tren\_ida, dmi.horario\_tren\_retor, dmi.nombre\_guia

FROM detalle\_machu\_itinerario dmi;

Archivo: rendimientoinicial.sql

EXPLAIN SELECT i.id\_itinerario, g.nombre, e.nombre\_estado

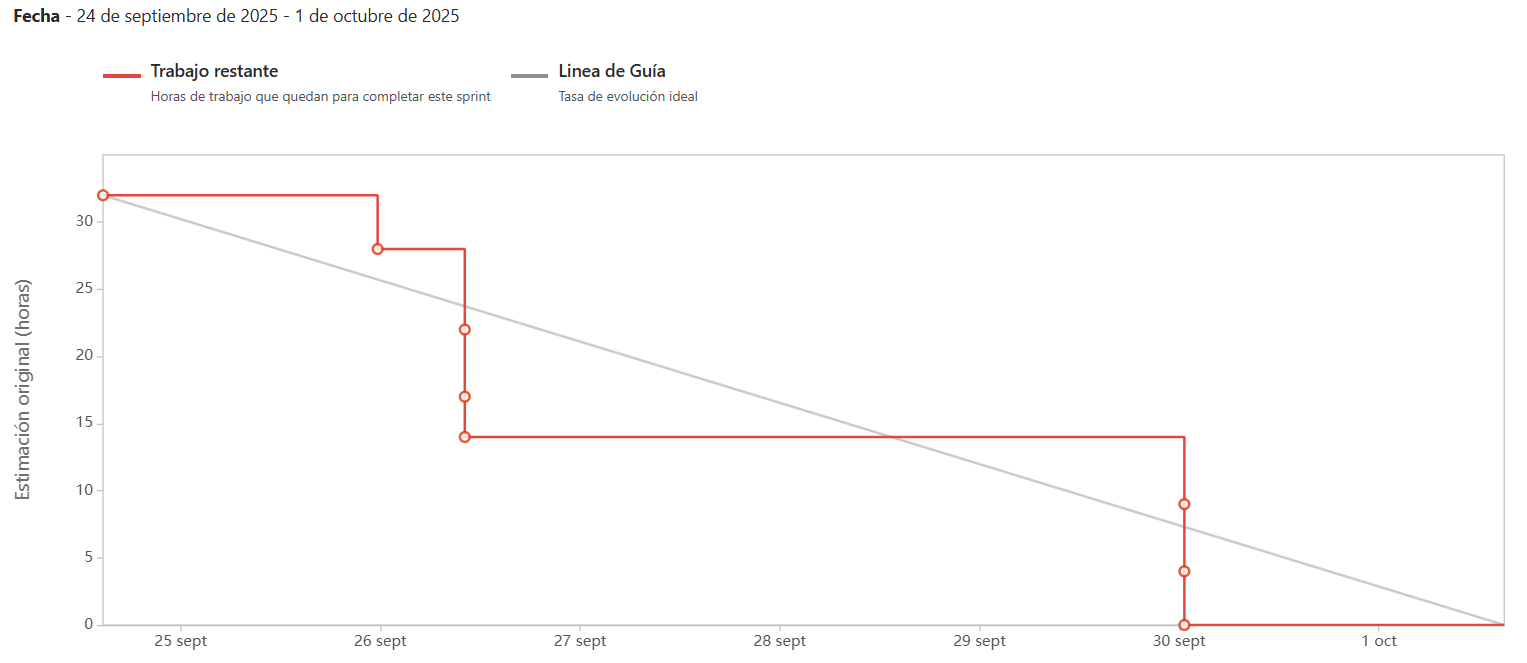
FROM itinerarios i

JOIN grupos g ON i.id\_grupo = g.id\_grupo

JOIN estados\_presupuesto e ON i.estado\_presupuesto\_id = e.id\_estado;

# 5. Burndown Chart

| **Backlog ID** | **Historia de Usuario** | **Tiempo Estimado** | **25/09/2025** | **26/09/2025** | **27/09/2025** | **28/09/2025** | **29/09/2025** | **30/09/2025** | **01/10/2025** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dia1** | **Dia2** | **Dia3** | **Dia4** | **Dia5** | **Dia6** | **Dia7** |
| **HU-19.1** | Configuración de respaldos automáticos | 18 | 4 | 11 | 3 |  |  |  |  |
| **HU-24.1** | Ejecución de prueba piloto |  |  |  |  | 5 | 6 | 9 |  |



# 6. Análisis del Sprint

* El trabajo se desarrolló de manera progresiva y ordenada, con una distribución de esfuerzo más equilibrada respecto al sprint anterior, lo que favoreció la continuidad del avance.
* Hubo errores al inicio relacionados con la configuración de respaldos, lo cual requirió investigar y probar soluciones; estos problemas quedaron documentados en el README para evitar que se repitan.
* Se fortaleció la documentación de los procesos de respaldo y validación, asegurando que el conocimiento quede disponible para futuras iteraciones y entornos de despliegue.
* El resultado final fue satisfactorio y se cumplieron las tareas estimadas.
* La carga de datos fue exitosa y se cumplieron todas las restricciones de integridad.
* La base de datos está lista para pruebas funcionales en un sistema real.

# 7. Conclusiones

La HU 19.1 se completó con éxito durante el Sprint 2. Aunque al inicio surgieron errores en la configuración de respaldos, se investigaron las posibles soluciones y se aplicaron de manera efectiva. Todo el proceso quedó documentado en el README para evitar que futuros desarrolladores enfrenten las mismas dificultades. Además, se avanzó en la documentación correspondiente, asegurando un mejor soporte para el mantenimiento y uso del sistema.

Con respecto a la las pruebas piloto se completaron y a su vez los resultados confirmaron la integridad de los datos. En comparación con el anterior sprint, se observó una mejor continuidad en la ejecución de las tareas.