

	Documento Diseño de Servicios	Página: 1 de 22
		Versión: 1
SYSREST - Sistema de Turnos		Fecha: 16/09/24
Transformación digital		

Middleware Notificaciones
SYSREST - Sistema de Turnos
Documento Diseño de Servicios:
Versión 3.0

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Revisión

Fecha	Versión	Descripción	Autor
16 - 09 - 2024	1.0	Primera especificación del Servicio para notificaciones para el software para turnos	Mateo Arias Cáceres
	2.0		
	3.0		

Aprobaciones

Nombre / Rol	Fecha	Firma

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Tabla de Contenido

1.	Breve Descripción	4
2.	Versión del servicio	4
3.	Dueños del servicio	4
4.	Descripción de la interfaz de los servicios	4
5.	Detalle de las funciones del servicio	5
5.1	createAppointment	5
5.2	getAppointment	8
5.3	updateAppointment	9
5.4	deleteAppointment	11
6.	Manejo de requerimientos no funcionales	13
7.	Manejo de requerimientos funcionales	13
7.1	Configuración en base de datos.	14
7.1.1	Tabla Clientes	14
7.1.2	Tabla Módulos	15
7.1.3	Tabla TiposServicio	15
7.1.4	Tabla tiposturno	16
7.1.5	Restricciones de la Tabla tiposturno	16
7.1.6	Descripción de Errores Relacionados	16
7.1.7	Tabla TiposTurno	17
7.1.8	Tabla TurnosAgendados	17
7.1.9	Tabla Usuarios	18
7.1.10	Relaciones en la Base de Datos:	19
8.	Manejo de Requerimientos No Funcionales	21
8.1	Rendimiento	21
8.2	Escalabilidad	21
8.3	Seguridad	21
9.	Manejo de Requerimientos Funcionales	21
9.1	Configuración en Base de Datos	21
9.2	Relaciones Entre Entidades	21
9.3	Lógica de Negocio	21
10.	Llamadas Asíncronas y Síncronas del Servicio	21
10.1	Llamadas Síncronas	21
10.2	Llamadas Asíncronas	21

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Documento de Especificación del Servicio de Entidad Incidente

1. Breve Descripción

SYSREST es un sistema web para gestionar el agendamiento de turnos en una organización. Permite a los usuarios solicitar turnos y recibir notificaciones automáticas por correo electrónico o SMS, asegurando la disponibilidad y auditoría de cada transacción.

2. Versión del servicio

1.0

3. Dueños del servicio

De acuerdo con una matriz RACI (Responsible, Accountable, Consulted, Informed) se indican las áreas de negocio o personas responsables del servicio:

Responsible	Mateo Arias Cáceres, Sara Iizeth Lozano, Wilmer Hernán Origua Reyes
Accountable	
Informed	

4. Descripción de la interfaz de los servicios

En esta sección se describen las funciones u operaciones que se pueden implementar usando los servicios entregados por el servicio:

url: <https://host:port/sysrest/v1/>

Nombre de la función o servicio	Descripción de la función	Información de entrada	Información de salida	Método HTTP
createAppointment	Crear un nuevo turno	cliente_ID, tipo_servicio_ID, tipo_turno_ID, fecha, hora, email, telefono.	Confirmación del turno creado, incluyendo el turno_ID.	POST
getAppointment	Obtener los detalles de un turno existente.	turno_ID.	Detalles del turno: cliente_ID, tipo_servicio, fecha, hora, estado.	GET
updateAppointment	Actualizar los detalles de un turno.	turno_ID, fecha, hora, estado.	Confirmación de la actualización o mensaje de error si no se encuentra el turno.	PUT
deleteAppointment	Cancelar o eliminar un turno existente.	turno_ID.	Confirmación de la cancelación del turno.	DELETE

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

getAvailableSlots	Obtener los horarios disponibles para un tipo de servicio.	tipo_servicio_ID, fecha.	Lista de horarios disponibles para la fecha seleccionada	GET
auditAppointments	Obtener un reporte de las transacciones de turnos (agendados/cancelados).	fecha_inicio, fecha_fin.	Lista de transacciones auditadas: turno_ID, cliente_ID, estado, fecha.	GET

5. Detalle de las funciones del servicio

En esta sección se describe el detalle las funciones y las reglas que aplican para cada uno de ellas que componen los mensajes de entrada y salida.

5.1 createAppointment

Función que permite crear un nuevo turno.

Información de entrada:			
Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
cliente_ID	Identificador del cliente	Integer	Número entero
tipo_servicio_ID	Identificador del servicio solicitado	Integer	Valores permitidos: 1 (Caja), 2 (Asesoría)
tipo_turno_ID	Identificador del tipo de turno (General o Preferencial)	Integer	1 (General), 2 (Preferencial)
fecha	Fecha del turno	DateTime	Formato ISO 8601 (YYYY-MM-DD)
hora	Hora del turno	Time	Formato ISO 8601 (HH:MM:SS)
email	Correo electrónico del cliente	String	Dirección válida de correo electrónico
telefono	Número de teléfono del cliente	String	Formato E.164 (Ejemplo: +573234557473)

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Descripción:

Esta función permite crear un nuevo turno en el sistema SYSREST. Un cliente puede solicitar un turno para un tipo de servicio y recibir una confirmación de forma inmediata.

Ejemplo REQUEST

```
{
  "cliente_ID": 101,
  "tipo_servicio_ID": 1,
  "tipo_turno_ID": 2,
  "fecha": "2024-09-15",
  "hora": "10:30:00",
  "email": "carlos.marin@gmail.com",
  "telefono": "+573234557473"
}
```

Información de salida:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
turno_ID	Identificador único del turno creado	Integer	Número entero
success	Indica si la creación fue exitosa	Boolean	true / false
message	Mensaje de confirmación o error	String	Mensaje de estado

Ejemplo response:

```
{
  "turno_ID": 12345,
  "success": true,
  "message": "Turno registrado correctamente"
}
```

Si se presenta algún error se pueden definir de dos tipos:

Técnicos:

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

codError	descError	Descripción de error
404	Not Found	URL inválida o recurso no encontrado.
422	Unprocessable Entity	La estructura del Request no valida.
500	Internal Server Error	Error interno del servidor.
405	Method Not Allowed	Método HTTP incorrecto (ejemplo: uso de GET en lugar de POST).

Formato response para Errores técnicos:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "422",
    "description": "Unprocessable Entity"
  }
}
```

Negocio:

codError	descError	codMessage	descMessage
201	Business Validation	NTF01	Cliente no encontrado en el sistema.
201	Business Validation	NTF02	El servicio solicitado no está disponible.
201	Business Validation	NTF03	Tipo de turno no permitido para el servicio.
201	Business Validation	NTF04	La fecha y hora seleccionadas no están disponibles.

Formato Response para errores de negocio:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "201",
    "description": "Business Validation",
    "codMessage": "NTF03",
    "descMessage": "Tipo de turno no permitido para el servicio."
  }
}
```

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

5.2 getAppointment

Función de obtener los datos de turnos existentes.

Información de entrada:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
turno_ID	Identificador del turno	Integer	Número entero

Descripción:

Esta función permite obtener los detalles de un turno existente en el sistema SYSREST utilizando el identificador del turno.

Ejemplo REQUEST

```
{
  "turno_ID": 12345
}
```

Información de salida:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
cliente_ID	Identificador del cliente	Integer	Número entero
tipo_servicio	Nombre del servicio solicitado	String	Ej: "Caja", "Asesoría"
fecha	Fecha del turno	DateTime	Formato ISO 8601 (YYYY-MM-DD)
hora	Hora del turno	Time	Formato ISO 8601 (HH:MM:SS)
estado	Estado del turno (ej: pendiente, confirmado)	String	Ej: "pending", "confirmed", "canceled"

Ejemplo response:

```
{
  "cliente_ID": 101,
  "tipo_servicio": "Caja",
  "fecha": "2024-09-15",
  "hora": "10:30:00",
  "estado": "confirmed"
}
```


Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Si se presenta algún error se pueden definir de dos tipos:

Técnicos:

codError	descError	Descripción de error
404	Not Found	URL o turno no encontrado.
422	Unprocessable Entity	Estructura de la solicitud no válida.
500	Internal Server Error	Error interno del servidor.

Formato response para Errores técnicos:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "404",
    "description": "Not Found"
  }
}
```

Negocio:

codError	descError	codMessage	descMessage
201	Business Validation	NTF05	El turno solicitado no existe en el sistema.

Formato Response para errores de negocio:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "201",
    "description": "Business Validation",
    "codMessage": "NTF05",
    "descMessage": "El turno solicitado no existe en el sistema."
  }
}
```

5.3 updateAppointment

Función actualizar los detalles de un turno.

Información de entrada:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
turno_ID	Identificador del turno	Integer	Número entero

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

fecha	Nueva fecha del turno	DateTime	Formato ISO 8601 (YYYY-MM-DD)
hora	Nueva hora del turno	Time	Formato ISO 8601 (HH:MM:SS)
estado	Nuevo estado del turno	String	"pending", "confirmed", "canceled"

Descripción:

Esta función permite actualizar los detalles de un turno existente en el sistema, como la fecha, hora o estado.

Ejemplo REQUEST

```
{
  "turno_ID": 12345,
  "fecha": "2024-09-16",
  "hora": "11:00:00",
  "estado": "confirmed"
}
```

Información de salida:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
success	Indica si la actualización fue exitosa	Boolean	true / false
message	Mensaje de confirmación o error	String	Mensaje de estado

Ejemplo response:

```
{
  "success": true,
  "message": "Turno actualizado correctamente"
}
```

Si se presenta algún error se pueden definir de dos tipos:

Técnicos:

codError	descError	Descripción de error
404	Not Found	URL o turno no encontrado.
422	Unprocessable Entity	Estructura de la solicitud no válida.
500	Internal Server Error	Error interno del servidor.

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Formato response para Errores técnicos:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "404",
    "description": "Not Found"
  }
}
```

Negocio:

codError	descError	codMessage	descMessage
201	Business Validation	NTF06	No se puede actualizar el turno con el estado proporcionado.

Formato Response para errores de negocio:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "201",
    "description": "Business Validation",
    "codMessage": "NTF06",
    "descMessage": "No se puede actualizar el turno con el estado proporcionado."
  }
}
```

5.4 deleteAppointment

Función cancelar o eliminar un turno existente.

Información de entrada:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
turno_ID	Identificador del turno	Integer	Número entero

Descripción:

Esta función permite cancelar o eliminar un turno existente.

Ejemplo REQUEST

```
{
  "turno_ID": 12345
}
```

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

Información de salida:

Tag@Atributo	Descripción	Tipo	Valores (Dominio, Valor por default, etc.)
success	Indica si la actualización fue exitosa	Boolean	true / false
message	Mensaje de confirmación o error	String	Mensaje de estado

Ejemplo response:

```
{
  "success": true,
  "message": "Turno cancelado correctamente"
}
```

Si se presenta algún error se pueden definir de dos tipos:

Técnicos:

codError	descError	Descripción de error
404	Not Found	Turno no encontrado.
500	Internal Server Error	Error interno del servidor.

Formato response para Errores técnicos:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
    "status": "404",
    "description": "Not Found"
  }
}
```

Negocio:

codError	descError	codMessage	descMessage
201	Business Validation	NTF07	No se puede cancelar un turno confirmado o finalizado.

Formato Response para errores de negocio:

```
{
  "ContextResponseType": null,
  "notification": null,
  "statusResponse": {
```

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

```

    "status": "201",
    "description": "Business Validation",
    "codMessage": "NTF07",
    "descMessage": "No se puede cancelar un turno confirmado o finalizado."
  }
}

```

6. Manejo de requerimientos no funcionales

En esta sección se describen los requerimientos no funcionales que deben soportarse alrededor del servicio:

Aspecto no funcional	Requerimiento	Detalle del requerimiento
Seguridad	Autenticación y autorización	El sistema debe utilizar autenticación basada en tokens (JWT) para validar a los usuarios que acceden a la API. Solo usuarios autenticados pueden crear, modificar o cancelar turnos.
	Encriptación de datos	La información sensible (correos, teléfonos, etc.) debe ser encriptada durante la transmisión utilizando HTTPS (SSL/TLS).
Disponibilidad	Alta disponibilidad	El servicio debe estar disponible 99.9% del tiempo. Esto implica la capacidad de recuperación ante fallos y redundancia del servicio.
	Escalabilidad	El sistema debe poder escalar horizontalmente para soportar un aumento en la demanda de solicitudes concurrentes.
Rendimiento	Tiempos de respuesta	El tiempo de respuesta máximo para la creación de un turno no debe exceder los 2 segundos en condiciones normales de operación.
Capacidad de procesamiento	Soporte de carga concurrente	El sistema debe soportar al menos 500 solicitudes simultáneas sin degradación del rendimiento.
Mantenibilidad	Código modular y documentado	El código debe ser mantenible, modular y seguir los estándares de desarrollo. Además, debe contar con documentación clara para su futuro mantenimiento.
Soporte y monitoreo	Monitoreo continuo	El sistema debe integrar herramientas de monitoreo (como Prometheus o Grafana) para observar el rendimiento, tiempos de respuesta y disponibilidad.
	Alertas de fallos	El sistema debe generar alertas automáticas a través de correo electrónico o sistemas de notificación en caso de fallos o caídas del servicio.
Seguridad de datos	Protección de datos personales	Debe cumplir con las normativas de protección de datos (ej: GDPR o la ley local) asegurando que la información personal esté protegida y solo accesible para personal autorizado.

7. Manejo de requerimientos funcionales

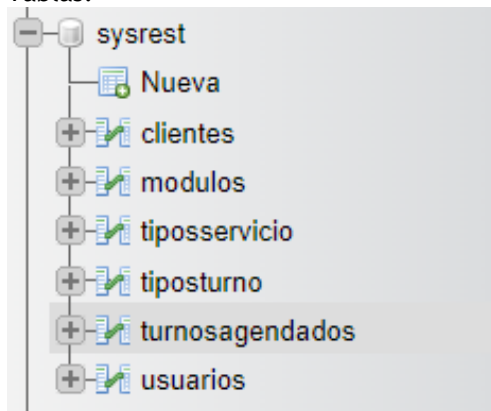
Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

7.1 Configuración en base de datos.

Motor de base de datos: MariaDB

Esquema: sysrest

Tablas:



7.1.1 Tabla Clientes

Esta tabla almacena la información de los clientes que utilizan los servicios de agendamiento de turnos.

Campo	Descripción	Tipo
cliente_id	Identificador único del cliente	INT (PK)
nombre_completo	Nombre completo del cliente	VARCHAR(50)
tipo_documento	Tipo de documento del cliente	VARCHAR(20)
numero_documento	Número de documento del cliente	VARCHAR(20)
estado	Estado del cliente (Activo/Inactivo)	VARCHAR(20)

Registros de ejemplo:

cliente_id	nombre_completo	tipo_documento	numero_documento	estado
1	Juan Perez	CC	123456789	Activo
2	Maria Lopez	TI	987654321	Inactivo

Restricciones:

Primary Key: cliente_id.

Restricciones de servicio y errores:

Error Técnico: Si se intenta registrar un cliente con un cliente_id ya existente, se generará un error de clave primaria duplicada.

Error de Negocio: Si se intenta registrar un cliente con un número de documento ya registrado, el servicio devolverá un mensaje de error:

```

{
  "status": "201",
  "message": "El número de documento ya está registrado."
}
  
```

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

7.1.2 Tabla Módulos

Gestiona los módulos donde los clientes pueden ser atendidos.

Campo	Descripción	Tipo
modulo_id	ID único del modulo	INT (PK)
nombre	Nombre del modulo	VARCHAR(50)
estado	Estado del modulo	VARCHAR(20)

Registros de ejemplo:

modulo_id	nombre	estado
1	Módulo 1	Activo
2	Módulo 2	Activo
3	Módulo 3	Inactivo

Restricciones:

- **Primary Key:** modulo_id.

Restricciones de servicio y errores:

- **Error Técnico:** Si se intenta asignar un turno a un modulo_id inexistente, se devolverá un error de clave foránea.
- **Error de Negocio:** Si el módulo solicitado no está disponible, el servicio devolverá el siguiente mensaje:

```
{
  "status": "201",
  "message": "El módulo solicitado no está disponible."
}
```

7.1.3 Tabla TiposServicio

Esta tabla define los tipos de servicios disponibles en el sistema para agendar turnos (ej.: Caja, Asesoría).

Campo	Descripción	Tipo
tipo_turno_id	Identificador único del tipo de servicio	INT (PK)
nombre_turno	Nombre del servicio (Ej.: "Caja", "Asesoría")	VARCHAR(50)
estado	Estado del servicio	VARCHAR(20)

Registros de ejemplo:

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

tipo_servicio_id	nombre_servicio	estado
1	Caja	Activo
2	Asesoría	Activo
3	Otros Servicios	Inactivo

Restricciones:

- **Primary Key:** tipo_servicio_id.

Restricciones de servicio y errores:

- **Error Técnico:** Si se intenta utilizar un tipo_servicio_id no registrado, se generará un error de clave foránea.
- **Error de Negocio:** Si el tipo de servicio solicitado no está disponible, el servicio devolverá el siguiente mensaje:

```
{
  "status": "201",
  "message": "El tipo de servicio solicitado no está disponible."
}
```

7.1.4 Tabla tiposturno

Permite la gestión y seguimiento de turnos agendados por los clientes.

Campo	Descripción	Tipo	Restricción
tipo_turno_id	Identificador único del tipo de turno	INT	Primary Key (PK)
nombre_turno	Nombre del tipo de turno (General, Preferencial)	VARCHAR(20)	

Registros de ejemplo:

tipo_turno_id	nombre_turno
1	General
2	Preferencial

7.1.5 Restricciones de la Tabla tiposturno

- **Primary Key:** tipo_turno_id.

7.1.6 Descripción de Errores Relacionados

- **Error Técnico:** Si se intenta usar un tipo_turno_id que no existe en la tabla tiposturno (por ejemplo, al referenciarlo desde otra tabla como turnosagendados), se generará un error de clave foránea.

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

- **Error de Negocio:** Si el tipo de turno solicitado no es válido (es decir, si no se encuentra en los registros de `tiposturno`), el sistema puede devolver un mensaje de error similar a:

```
{
  "status": "201",
  "message": "El tipo de turno solicitado no es válido."
}
```

7.1.7 Tabla TiposTurno

Esta tabla contiene los tipos de turnos que se pueden solicitar en el sistema (Ej.: General, Preferencial).

Campo	Descripción	Tipo
tipo_turno_id	Identificador único del tipo de turno	INT (PK)
nombre_turno	Nombre del turno (Ej.: "General", "Preferencial")	VARCHAR(45)

Registros de ejemplo:

tipo_turno_id	nombre_turno
1	General
2	Preferencial
3	General

Restricciones:

- **Primary Key:** `tipo_turno_id`.

Restricciones de servicio y errores:

- **Error Técnico:** Si se intenta agendar un turno con un tipo de turno no registrado, se generará un error de clave foránea.
- **Error de Negocio:** Si el tipo de turno solicitado no es válido, el servicio devolverá un mensaje de error:

```
{
  "status": "201",
  "message": "El tipo de turno solicitado no es válido."
}
```

7.1.8 Tabla TurnosAgendados

Esta tabla almacena la información sobre los turnos agendados por los clientes, incluyendo el tipo de servicio, el tipo de turno, la fecha, la hora y el estado del turno.

Campo	Descripción	Tipo
turno_id	Identificador único del turno agendado	INT (PK)

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

cliente_id	Relación con el cliente que agendó el turno	INT (FK)
tipo_servicio_id	Relación con el tipo de servicio solicitado	INT (FK)
tipo_turno_id	Relación con el tipo de turno solicitado	INT (FK)
fecha	Fecha del turno agendado	DATE
hora	Hora del turno agendado	TIME
estado	Estado actual del turno (Ej.: "pending", "confirmed", "canceled")	VARCHAR(20)

Registro de ejemplo:

turno_id	cliente_id	tipo_servicio_id	tipo_turno_id	fecha	hora	estado
1	1	1	1	15/04/2024	10:00:00	confirmed
2	2	2	2	16/04/2024	14:30:00	pending
3	1	3	3	17/04/2024	16:00:00	canceled

Restricciones:

- **Primary Key:** turno_id.
- **Foreign Keys:**
 - cliente_id referencia a Clientes(cliente_id).
 - tipo_servicio_id referencia a TiposServicio(tipo_turno_id).
 - tipo_turno_id referencia a TiposTurno(tipo_turno_id).

Restricciones de servicio y errores:

- **Error Técnico:** Si el cliente, tipo de servicio o tipo de turno referenciados no existen, se generará un error de restricción de clave foránea.
- **Error de Negocio:** Si se intenta agendar un turno con un estado no permitido, el servicio mostrará:

```
{
  "status": "201",
  "message": "El estado del turno no es válido."
}
```

7.1.9 Tabla Usuarios

Gestiona la información de los usuarios que interactúan con el sistema.

Campo	Descripción	Tipo
usuario_id	ID único del usuario	INT (PK)
nombre_usuario	Nombre de usuario para autenticación	VARCHAR(50)
contraseña	Contraseña encriptada del usuario	VARCHAR(255)

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

modulo_id	Módulo asignado al usuario	INT (FK)
tipo_servicio_id	Tipo de servicio que maneja el usuario	INT (FK)

Registros de ejemplo:

usuario_id	nombre_usuario	contraseña	modulo_id	tipo_servicio_id
1	Jose Garcia	\$2y\$10\$3SAIEY/O...	1	1
2	Martha Lopez	\$2y\$10\$hEiySqmn...	2	2

Restricciones:

- **Primary Key:** usuario_id.
- **Foreign Keys:** modulo_id, tipo_servicio_id.

Restricciones de servicio y errores:

- **Error Técnico:** Si se intenta asignar un usuario a un modulo_id o tipo_servicio_id que no existen, se generará un error de clave foránea.
- **Error de Negocio:** Si el nombre de usuario ya está registrado, el servicio devolverá el siguiente mensaje:

```
{
  "status": "201",
  "message": "El nombre de usuario ya esta en uso."
}
```

7.1.10 Relaciones en la Base de Datos:

Con base en la estructura de la base de datos que proporcionaste, las relaciones entre las tablas se describen de la siguiente manera:

7.1.10.1 Clientes -> TurnosAgendados

Un cliente puede tener múltiples turnos agendados.

Descripción de la Relación:

- La tabla clientes tiene una relación **uno a muchos** con la tabla turnosagendados, lo que significa que un cliente (cliente_id) puede agendar varios turnos en el sistema, pero cada turno solo pertenece a un cliente.

Implementación Técnica:

- La columna cliente_id en la tabla turnosagendados es una **clave foránea** que referencia la columna cliente_id de la tabla clientes.

SQL de la Relación:

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

```
ALTER TABLE `turnosagendados`
ADD CONSTRAINT `fk_turnosagendados_clientes`
FOREIGN KEY (`cliente_id`) REFERENCES `clientes` (`cliente_id`);
```

7.1.10.2 TiposServicio -> TurnosAgendados

Un tipo de servicio puede estar asociado a varios turnos agendados.

Descripción de la Relación:

- La tabla tiposservicio tiene una relación **uno a muchos** con la tabla turnosagendados, lo que significa que un tipo de servicio (tipo_servicio_id) puede estar asociado a varios turnos, pero cada turno solo puede tener un tipo de servicio.

Implementación Técnica:

- La columna tipo_servicio_id en la tabla turnosagendados es una **clave foránea** que referencia la columna tipo_servicio_id de la tabla tiposservicio.

SQL de la Relación:

```
ALTER TABLE `turnosagendados`
ADD CONSTRAINT `fk_turnosagendados_tiposservicio`
FOREIGN KEY (`tipo_servicio_id`) REFERENCES `tiposservicio` (`tipo_servicio_id`);
```

7.1.10.3 TiposTurno -> TurnosAgendados

Un tipo de turno puede estar relacionado con varios turnos agendados.

Descripción de la Relación:

- La tabla tiposturno tiene una relación **uno a muchos** con la tabla turnosagendados, lo que significa que un tipo de turno (tipo_turno_id) puede aplicarse a varios turnos, pero cada turno solo tiene un tipo de turno.

Implementación Técnica:

- La columna tipo_turno_id en la tabla turnosagendados es una **clave foránea** que referencia la columna tipo_turno_id de la tabla tiposturno.

SQL de la Relación:

```
ALTER TABLE `turnosagendados`
ADD CONSTRAINT `fk_turnosagendados_tiposturno`
FOREIGN KEY (`tipo_turno_id`) REFERENCES `tiposturno` (`tipo_turno_id`);
```

Estas relaciones establecen cómo las entidades clave del sistema (clientes, tipos de servicio y tipos de turno) interactúan con los turnos agendados. De esta manera, garantizamos la integridad referencial de la base de datos, asegurando que todos los turnos agendados correspondan a entidades válidas.

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

8. Manejo de Requerimientos No Funcionales

El sistema Sysrest tiene que cumplir con ciertos requisitos no funcionales para asegurar su rendimiento, escalabilidad, y seguridad. Entre estos requerimientos, se destacan los siguientes:

8.1 Rendimiento

El sistema debe ser capaz de procesar solicitudes simultáneas sin degradar el rendimiento. El tiempo máximo de respuesta aceptable es de 2 segundos por cada solicitud de turno, independientemente del número de usuarios concurrentes.

8.2 Escalabilidad

Sysrest está diseñado para escalar vertical y horizontalmente. Se prevé que el sistema será capaz de manejar cientos de módulos y miles de usuarios simultáneamente, sin que ello afecte su rendimiento.

8.3 Seguridad

El sistema debe garantizar la seguridad de los datos mediante la autenticación de usuarios con contraseñas encriptadas y la implementación de tokens de sesión para mantener las interacciones seguras. El acceso a los módulos y tipos de servicio estará controlado mediante roles asignados a cada usuario.

9. Manejo de Requerimientos Funcionales

9.1 Configuración en Base de Datos

La configuración de la base de datos de Sysrest incluye varias tablas que interactúan entre sí para asegurar el funcionamiento adecuado del sistema. Estas tablas incluyen 'clientes', 'turnos agendados', 'usuarios', 'modulos', entre otras.

9.2 Relaciones Entre Entidades

Las entidades dentro de Sysrest están altamente interrelacionadas para garantizar la integridad de los datos. A continuación, se detallan algunas relaciones clave entre las entidades:

- ****Clientes -> Turnos Agendados****: Un cliente puede tener múltiples turnos asignados, cada uno asociado a un módulo de atención.
- ****Módulos -> Turnos Agendados****: Un turno puede estar asociado a un solo módulo en un momento dado.
- ****Usuarios -> Módulos****: Un usuario puede estar asignado a un módulo específico y puede gestionar varios turnos durante su sesión.

9.3 Lógica de Negocio

La lógica de negocio dentro de Sysrest asegura que los turnos se asignen de manera eficiente y que los usuarios puedan gestionar los módulos de atención sin inconvenientes. Algunos de los estados más importantes dentro de la lógica de negocio son:

- ****Turno En Proceso****: Indica que el cliente está siendo atendido en ese momento.
- ****Turno Atendido****: Señala que el cliente ha sido atendido y que el turno está cerrado.
- ****Turno Cancelado****: Indica que el cliente canceló el turno o que no se presentó.

10. Llamadas Asíncronas y Síncronas del Servicio

Sysrest soporta dos tipos de llamadas para procesar las solicitudes de turno: asíncronas y síncronas.

10.1 Llamadas Síncronas

En las llamadas síncronas, el sistema procesa las solicitudes de turno de manera inmediata y devuelve el estado final del turno una vez que la solicitud ha sido completamente procesada. Esto es útil para situaciones donde es necesario que el cliente reciba una respuesta inmediata.

10.2 Llamadas Asíncronas

En las llamadas asíncronas, el sistema recibe la solicitud de turno y responde de inmediato con un

Api Notificaciones	Versión: 3.0
Documento de especificación de servicios	Fecha: 16/07/20

mensaje de aceptación, procesando el turno en segundo plano. Esto es ideal para situaciones donde el procesamiento puede tardar más tiempo, pero el cliente no necesita una respuesta inmediata.