# Sistema de Gestión de turnos de empleados (Sis-GTE)

Breve Descripción:

Sistema que tiene por objetivo principal el de asignar turnos horarios a Empleados. Los turnos o frecuencias horarias asignadas deben están previamente definidas en el Cliente.

A los empleados se les podrán asignar “n” frecuencias de destinos clientes.

El usuario del sistema podrá contemplar dichos horarios en la visualización de un mes calendario (es decir las frecuencias asignadas repetidas en cuatro semanas) y podrán habilitarse o deshabilitarse (por ej. en caso de ausencia al servicio) lo que implicaría un reducción de la carga horario mensual.

# Servicios Backend Curso Full Stack

Aclaración preliminar sobre las rutas de tipo **delete y read**. De todas las rutas presentadas aquí como Principales se consideran que por la naturaleza del curso son innecesarias ya que podrían realizar el mismo objetivo en las rutas **update e index**.

Por Ejemplo:

**userRoutes.put('/delete)** // ruta para actualizar el estado del usuario a inhabilitado ( en el update tranquilamente se puede marcar el estado (activo o inactivo)

**userRoutes.put('/read)** // ruta para mostrar +Info del Usuario (para mostrar más información solo se podría ampliar el objeto en un modal por ej. Mostrando el resto de datos no mostrados en la tabla.

## Principales

1 – Usuarios Ruta Aplicación en index.ts**: server.app.use('/users', userRoutes);**

**userRoutes.post('/login')** // ruta para autenticarme en la aplicación

**userRoutes.get('/index)** // ruta para mostrar una tabla con el listado de usuarios

**userRoutes.post('/create)** // ruta para almacenar un nuevo usuario

**userRoutes.put('/update)** // ruta para actualizar datos del usuario

**userRoutes.put('/updatePassword)** // ruta para cambiar la password

2 – Empleados Ruta Aplicación en index.ts**: server.app.use('/empleados, empleadoRoutes);**

**empleadoRoutes.get('/index)** // ruta para mostrar una tabla con el listado de empleados

**empleadoRoutes.post('/create)** // ruta para almacenar un nuevo empleado

**empleadoRoutes.put('/update)** // ruta para actualizar datos del empleado

3 – Clientes Ruta Aplicación en index.ts**: server.app.use('/clientes, clienteRoutes);**

**clienteRoutes.get('/index)** // ruta para mostrar una tabla con el listado de clientes

**clienteRoutes.post('/create)** // ruta para almacenar un nuevo cliente

**clienteRoutes.put('/update)** // ruta para actualizar datos del cliente

4 – Servicios Ruta Aplicación en index.ts**: server.app.use('/servicios, servicioRoutes);**

**servicioRoutes.post('/create)** // ruta para almacenar un nuevo servicio correspondiente a un Cliente. El usuario seteará para cada día de la semana el horario de inicio y salida del servicio. ( se repetirá por defecto los mismos horarios en la totalidad del mes)

**servicioRoutes.get('/verServiciosAsignables)** // ruta para Ver los servicios no asignados a ningún empleado.

**servicioRoutes.put('/asignarServicioAempleado)** // ruta que realiza un update en la tabla “frecuencia\_horaria” marcando el campo “asignado en true”. Solo podrá asignarse los servicios no asignados y que no estén dados de baja. (EN ESTE METODO TAMBIEN SE ENVIA UN CORREO AUTOMATICO AL CLIENTE INFORMANDO LOS DATOS DEL NUEVO EMPLEADO QUE REALIZARA EL SERVICIO)

**servicioRoutes.put('/desasignarServicioAempleado)** // ruta que realiza un update en la tabla “frecuencia\_horaria” marcando el campo “asignado en false”.

**servicioRoutes.get('/serviciosDelEmpleado)** // ruta para acceder a los servicios asignados al empleado, se mostrara en una tabla los clientes que le fueron asignados con la suma de las respectivas frecuencias de horarios semanales.

**servicioRoutes.put('/delete)** // ruta para hacer un update en el estado de la frecuencia en que se dará de baja, se completa el campo deleted\_at con la fecha actual.

**servicioRoutes.get('/getMesCalendario)** // ruta que retorna el registro completo de la frecuencia horario de un determinado empleado. Por cuestiones de legibilidad se divide el registro en 4 semanas. Ademas retorna el cálculo del total de horas trabajadas

OPCIONALES

**servicioRoutes.put('/cerrarMesActual)** // ruta que realiza un update en la tabla “frecuencia\_horaria” dejando todo todos los días del mes activos, Solo podrá asignarse los servicios no asignados y que no estén dados de baja. (OPCIONAL para aprobar el curso no lo hacemos)

**servicioRoutes.put('/uptateFrecuencia)** // ruta que realiza un update en la tabla “frecuencia\_horaria”, nos devuelve la frecuencia Seleccionada con la semana correspondiente con todos los horarios de inicio y fin de esa semana para ser modificados , aumentando o disminuyendo las horas totales del dia. (OPCIONAL para aprobar el curso no lo hacemos)

//////////////////////////////////////////////////\*\* OBS IMPORTANTES ( AL FINAL LO TERMINE HACIENDO)

Se replican la frecuencia definida en el cliente por 4 en la base de datos (representando un mes calendario). ESTO ES DE ACUERDO A LOS REQUERIMIENTOS DEL SISTEMA PERO SE VA A COMPLICAR HACERLO PARA EL CURSO OPINO NO HACERLO DE IGUAL MANERA ARMO LA BASE COMO PARA Q ESTO FUNCIONE

Horarios Semanales (la idea es en realidad mostrar los 30 días del mes pero nos vamos a complicar una banda me parece, capaz mostramos nada mas q el horario que hace en la semana) y q le calcule más o menos el tiempo que trabaja al mes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FIN OBS IMPORTANTES //////////////////

Ej.

Empleado1

Cumple Servicios en los siguientes clientes y en los horarios:

Semana 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cliente | Lunes | Martes | Miércoles | Jueves | Viernes | Sábado | Domingo | Opciones |
| Cliente1 | 7 a 11 |  | 7 a 11 |  | 7 a 11 |  |  | Amplia info Cliente/Modificar Horarios frecuencia |
| Cliente2 |  | 8 a 12 |  | 8 a 12 |  | 8 a 12 |  | Ampliar info Cliente |
| Total horas | 4 | 4 | 4 | 4 | 4 | 4 |  |  |

Luego se repiten por el total de 4 semanas.

\*(el precio de la hora sale de un campo perteneciente al empleado)

Trabaja al mes 96 hs a razón de $198 la hora. Total $19008.

5 – Dashboard Ruta Aplicación en index.ts**: server.app.use('/dashboard, dashboardRoutes);**

**dashboardRoutes.get('/getCantidadEmpleados)** // ruta que retorna la cantidad de empleados (Se pueden ver quienes pero no en esta versión)

**dashboardRoutes.get('/getCantidadEmpleadosSinServicios)** // ruta que retorna la cantidad de empleados que no tienen su frecuencia horario creada. . (Se pueden ver quienes pero no en esta versión)

**dashboardRoutes.get('/getCantidadClientes) //** ruta que retorna la cantidad de clientes (Se pueden ver quienes pero no en esta versión)

**dashboardRoutes.get('/getCantidadClientesSinServicios)** // ruta que retorna la cantidad de clientes que no tienen su frecuencia horario creada. . (Se pueden ver quienes pero no en esta versión)

**dashboardRoutes.get('/getTopEmpleadosConMenosHorasDeServicios)** // ruta que retorna los nombres de los 10 empleados con MENOS horas acumuladas de servicios. . (se pueden ver mas detalles pero no en esta versión) (Es una grafica)

**dashboardRoutes.get('/getTopEmpleadosConMasHorasdDeServicios)** // ruta que retorna los nombres de los 5 empleados con MAS cantidad de servicios. . (se pueden ver más detalles pero no en esta versión) (Es una grafica)

5 – Archivos Ruta Aplicación en index.ts**: server.app.use('/archivos, archivosRoutes);**

**Ruta para el storage de archivos e imágenes**

AHÍ TENDRIAMOS 6 SERVICIOS GRANDES E IMPORTANTES. RESTAN OTROS Q SON MAS DE LA PARTE DE ADMISTRACION O GESTION DEL SISTEMA.

Serian con rutas bien simples index, create y update

-Categoría IVA.

-Roles Usuarios

-Tipo Empleados

-Cálculos y Reportes y Estadísticas

-Caja