

# Requerimientos

El cliente plantea que desea un sistema para poder administrar las urgencias dentro del hospital. Quiere poder registrar los datos básicos del paciente, una descripción del estado y el nivel de urgencia asignado al paciente.

El cliente plantea que desea guardar los siguientes datos del paciente:

- Nombre
- Apellido
- Edad
- Descripción del estado (texto amplio para poder cargar información)
- Fecha de ingreso (cuando el paciente ingresa)
- Fecha de egreso (cargada solamente si el paciente salió de la urgencia)
- Nivel de urgencia (que se detalla a continuación)

Del nivel de urgencia, el cliente desea guardar:

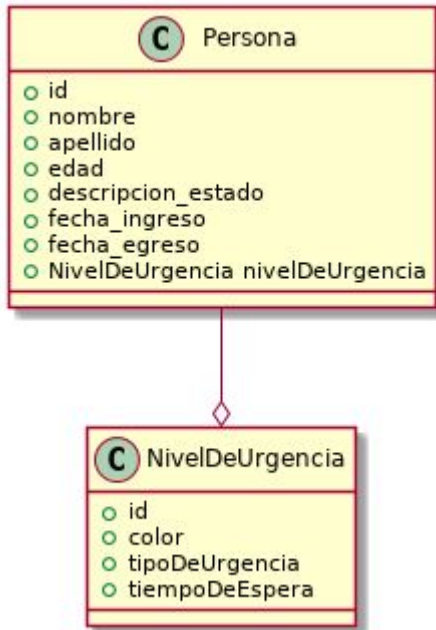
- Color
- Tipo de urgencia (un texto descriptivo)
- Tiempo de espera máximo (el tiempo máximo que el paciente debe esperar para ser atendido)

Actualmente los niveles de urgencia ya están establecidos (pero el cliente desea poder administrar los mismos en caso de ser necesario).

## Niveles de urgencia

Color	Tipo de urgencia	Tiempo de espera máximo
Rojo	Resucitación	0
Naranja	Emergencia	15 minutos
Amarillo	Urgencia	60 minutos
Verde	Urgencia menor	2 horas
Azul	Sin urgencia	4 horas

El equipo de diseño ha planteado una estructura de clases para que puedas tomar de base para el desarrollo



## Requisitos del sistema

- La implementación del sistema debe ser por medio de API REST
- Se deben poder administrar los niveles de urgencia (dar de alta, modificar, borrar lógicamente, listar, y ver uno)
- La ruta para realizar las operaciones sobre los niveles de urgencia debe ser /api/niveles\_de\_urgencia
- Se deben poder administrar las personas (dar de alta, modificar, borrar lógicamente, listar, y ver uno)
- El listado de personas, solo debe incluir aquellas personas que no hayan egresado (fecha\_egreso en null)
- La ruta para realizar las operaciones sobre las personas debe ser /api/personas
- Tanto el listado de personas, como la respuesta de una persona deben incluir la información del nivel de urgencia asignado a dicha persona
- No se deben borrar físicamente los registros, solo se permitirán borrados lógicos

Recuerda que debes solamente retornar las personas que no hayan egresado, por lo que deberás ver en la página de Laravel como puedes aplicar filtros

(<https://laravel.com/docs/6.x/eloquent>)

## Forma de trabajo

- Puedes realizar el trabajo en equipo de hasta 3 participantes
- Haz tu mejor esfuerzo para llegar con la fecha límite que nos planteó el cliente (sábado 16 de Noviembre)
- No debes entregar el TP, pero sí te recomendamos realizarlo para poder aprovechar al máximo el curso.

El sábado 16 de Noviembre a las 19hs estaremos resolviendo este ejercicio online.