

IES JOSÉ RODRIGO BOTET

PROYECTO DE DESARROLLO DE APLICACIONES WEB

**APLICACIÓN WEB PARA EL NOMBRAMIENTO DE TAREAS O SERVICIOS DE BASE/COMPAÑÍA**

|  |
| --- |
| NOMBRE: Germán Orlando Navarro Díaz  DNI: 50516754Z  NIA: 11395657  CURSO: 2018/2019  CONVOCATORIA: Ordinaria |

**TRIBUNAL**

Fdo. PRESIDENTE:

Fdo. TUTOR INDIVIDUAL:

Fdo.TUTOR COLECTIVO:

En Manises, a……. de.......................................... de 2019.

|  |
| --- |
| NOTA FINAL: |
|  |

# INDICE

Tabla de contenido

[INDICE 3](#_Toc10756826)

[INTRODUCCIÓN 5](#_Toc10756827)

[HERRAMIENTAS UTILIZADAS 6](#_Toc10756828)

[LENGUAJES EMPLEADOS. 6](#_Toc10756829)

[a. PHP (Hypertext Preprocessor) 6](#_Toc10756830)

[b. JavaScript 6](#_Toc10756831)

[c. HTML (HyperText Markup Language) 8](#_Toc10756832)

[d. CSS (Cascading Style Sheets) 8](#_Toc10756833)

[e. MySQL 9](#_Toc10756834)

[FRAMEWORKS 10](#_Toc10756835)

[f. Laravel (PHP) 10](#_Toc10756836)

[g. Bootstrap (CSS) 12](#_Toc10756837)

[LA BASE DE DATOS 12](#_Toc10756838)

[ANÁLISIS DE REQUISITOS 14](#_Toc10756839)

[PLANTEAMIENTO DE NECESIDADES 14](#_Toc10756840)

[QUIÉN PODRÁ HACER USO DE LA APLICACIÓN. CASOS DE USO 15](#_Toc10756841)

[LA APLICACIÓN WEB 17](#_Toc10756842)

[DESCRIPCIÓN 17](#_Toc10756843)

[CONSTITUCIÓN 17](#_Toc10756844)

[PLANTILLAS DE VISUALIZACIÓN 19](#_Toc10756845)

[ASIGNACIÓN DE TAREAS 20](#_Toc10756846)

[CONFIGURACIÓN 23](#_Toc10756847)

[CONCLUSIONES. 24](#_Toc10756848)

[TRABAJOS FUTUROS. 25](#_Toc10756849)

[BIBLIOGRAFÍA. 26](#_Toc10756850)

[ANEXOS 27](#_Toc10756851)

[Anexo 1 – Relaciones de la base de datos. 27](#_Toc10756852)

[Anexo 2 - Código para crear la tabla “tasks” usando la migración de Laravel 28](#_Toc10756853)

[MANUAL INSTALACIÓN/CONFIGURACIÓN/USO. 29](#_Toc10756854)

# INTRODUCCIÓN

#### NOMBRAMIENTO DE SERVICIOS DE BASE/COMPAÑÍA

Desde que existen las fuerzas militares en España ha existido la necesidad de asignar tareas cotidianas o de seguridad al personal militar para su realización. Tareas como la limpieza de espacios habitables, de vehículos, de armamento, etc, al igual que guardias de seguridad o guardias de orden. En un principio, probablemente, esas tareas eran asignadas por el jefe encargado, a su criterio, o por un subordinado con la autoridad para ello.

Conforme avanzan los tiempos se podría decir que  esta asignación de tareas ahora se lleva a cabo de un modo más metódico, siguiendo una serie de criterios y normas preestablecidas bajo un marco legal y  organizacional. Algunos de esos criterios a tener en cuenta son la antigüedad en el empleo; si la persona estará ausente el día que le corresponde hacer el servicio ya sea por causas médicas o por estar fuera de la base en un ejercicio, de maniobras, ejercicio de tiro; reducción de jornada, etc.

A fecha de hoy, el tiempo normalmente invertido por la persona o personas encargadas de la asignación de las tareas suele ser superior a una hora, pero cuando se aproximan festividades el tiempo empleado puede llegar a  duplicarse o triplicarse debido a la gran cantidad de variables a tener en cuenta.

 El objetivo principal de este proyecto consiste en desarrollar una herramienta que permita la asignación de dichas tareas empleando el menor tiempo posible. Para ello se propone el uso de una base de datos relacional, a la cual, a través de una página web, se le proporciona la información necesaria con el uso de tecnologías web como MYSQL, PHP, JavaScript, AJAX o JQUERY.

# HERRAMIENTAS UTILIZADAS

El desarrollo de esta aplicación se ha realizado haciendo uso de diferentes lenguajes de programación teniendo en cuenta que cada uno posee unascaracterísticas específicas que le dan valor gracias a las mismas. Unos son ejecutados en el cliente y otros en el servidor.

## LENGUAJES EMPLEADOS.

### PHP (Hypertext Preprocessor)

PHP es un lenguaje de script (instrucciones) que se ejecuta en el servidor. Es de código abierto (open source). Se utiliza para desarrollar páginas web estáticas o dinámicos, así como aplicaciones web. PHP es sinónimo de preprocesador de hipertexto, que anteriormente significaba páginas de inicio personales (Personal Home Page). Los scripts PHP solo se pueden interpretar en un servidor que tiene PHP instalado.

PHP es el lenguaje principal usado como base para toda esta aplicación web.

### JavaScript

Es un lenguaje de programación interpretado principalmenteusado en desarrollo web. Fue desarrollado con el propósito de dar dinamismo e interactividad a los elementos de las páginas web. Está basado en ECMAScript, otro lenguaje de programación desarrollado por Sun Microsistemas.

A diferencia de PHP, JavaScript se ejecuta en el cliente y no en el servidor. Esto significa que se pueden ejecutar funciones en el cliente una vez la página ha cargado sin necesidad de que exista comunicación con un servidor. Esto permite a las páginas reaccionar a eventos, exhibir efectos especiales, aceptar texto variable, validar datos, crear cookies, detectar el navegador de un usuario.

Al igual que PHP, JavaScript se puede insertar en el código HTML.

JavaScript fue usado en esta aplicación web para mostrar advertencias, usando Ajax, y JQuery.

#### AJAX (Asynchronous JavaScript And XML)

Siglas de: JavaScript y XML Asíncrono. Ajax no es un lenguaje de programación. Es una nueva técnica para crear mejores, más rápidas, y más interactivas las aplicaciones web con la ayuda de XML, JSON, HTML, CSS, y JavaScript.

Con Ajax cuando se envía un formulario JavaScript realiza la solicitud al servidor, interpreta los resultados, y actualiza la pantalla sin recargar la página actual necesariamente. Todo sucede en segundo plano, sin que el usuario lo note.

Entre las características de Ajax se encuentran:

* Actualizar una página web sin volver a cargar la página.
* Solicitar datos de un servidor - después de que la página se haya cargado
* Recibir datos de un servidor - después de que la página se haya cargado
* Enviar datos a un servidor - en segundo plano.

Ajax fue usado en esta aplicación web para solicitar al servidor un listado de los “empleos” asignados a cada tarea en el momento que se selecciona una tarea.

#### JQuery

Es una librería de Javascript cuyo principal propósito es simplificar la manipulación de: los elementos de una página web (DOM), las llamadas a servidor usando Ajax, CSS, y el manejo de eventos.

JQuery fue usado en esta aplicación web para modificar elementos del DOM agregando u ocultando botones, mostrando advertencias, haciendo peticiones con Ajax, etc.

#### DataTables

Es una librería que usa JQuery cuyo propósito es facilitar la manipulación, dar formato y añadir algunas características avanzadas a las tablas HTML.

### HTML (HyperText Markup Language)

Lenguaje de Marcas de Hipertexto. Utilizado para el desarrollo de páginas web. Los elementos HTML son la estructura más básica de una página web. Se representan con una etiqueta, por ejemplo: body, table. La mayoría de las etiquetas tiene una etiqueta inicial y una etiqueta de cierre, ej: <body> contenido </body>; algunas etiquetas no requieren etiqueta de cierre, ej: <br>, <hr>.

HTML fue usado en todas las páginas de esta aplicación web, sin excepción.

### CSS (Cascading Style Sheets)

Es un lenguaje declarativo que describe cómo se muestran las páginas web en un navegador. Se puede usar para definir estilos de texto, tamaño de tablas, y otros aspectos que antiguamente sólo se podían definir en el código HTML. El código CSS puede ser definido en un archivo aparte, esto permite que varias páginas puedan reutilizar el mismo código y evitar la repetición de las declaraciones.

CSS3 es una versión mejorada y ampliada de las versiones anteriores, mejoras como esquinas redondeadas, sombras, transiciones, animaciones, etc.

CSS3 es el lenguaje principal usado para dar estilo a toda esta aplicación web.

### MySQL

Es un sistema de administración de bases de datos relacionales basado en SQL (Structure Query Language, o Lenguaje de Consultas Estructurado). Su aplicación es altamente diversificada. Sin embargo, la aplicación más común de MySQLse da en las base de datos web. Puede usarse para almacenar desde un simple dato (contador de vistitas) hasta un gran inventario de productos de una tienda online.

Usado en conjunto con lenguajes como PHP es posible crear sitios web que interactúan en tiempo real con dichas bases de datos.

## FRAMEWORKS

Antes de empezar, qué es un **Framework**? Se podría decir que un Framework es un conjunto de estructuras y guías utilizados para construir algo útil y con sentido.

Qué es un **Web Framework**? Es un software que ayuda a construir servicios de aplicaciones web y APIs. Permite la reutilización de componentes creados para llevar a cabo tareas repetitivas de modo más fácil.

Qué es un **Framework PHP?** Es un conjunto de clases PHP que ayudan a construir una aplicación web con facilidad.

### Laravel (PHP)

Laravel es un Framework PHP empleado en el desarrollo de aplicaciones web. Sigue la estructura MVC (Modelo Vista Controlador), que hace que sea más rápido crear estructuras web ya que posee funciones como autenticación, servicio de correo, enrutamiento, sesiones, etc.

Gracias a la facilidad para personalizar proyectos el tiempo empleado suele ser más reducido que siguiendo métodos tradicionales de programación

Entre las características de Laravel se encuentran:

* **Modularidad**: Los componentes creados pueden ser separados y reunificados para hacer la aplicación funcional.
* **Autenticación**: Tiene un sistema de autenticación que se puede implementar con la aplicación de una instrucción.
* **Caché temporal**: Crea una propia cache temporal para facilitar la reutilización de los datos usados con más frecuencia.
* **Enrutamiento**: permite agrupar, nombrar, filtrar o vincular rutas a los modelos fácilmente, además de usar rutas amigables.
* **Plantilla de visualización**: **Blade** es plantilla de visualización de Laravel. Posee funciones que ayudan a dar formato a los datos dentro de las vistas. Además, implementa la *herencia,* lo que permite usar *estructuras base* (layouts) que pueden ser reutilizadas con facilidad. Todas las vistas de blade tienen la extensión .blade.php
* **Constructor de consultas:** sin duda una herramienta muy valiosa. Permite realizar consultas con mucha facilidad de un modo legible para el usuario. También es posible realizar consultas del modo tradicional si fuese necesario.
* **ORM (Object-Relation Mapping) Eloquent**: Mapeo de Objetos y relaciones, algo así como **“**qué objeto está relacionado con cuál” en la base de datos. Laravel posee una forma muy sencilla de relacionar los objetos a través de los modelos creados.
* **Sistema de migraciones:** Se pueden crear tablas, bases de datos, índices, usando el sistema de migraciones. Es posible modificar tablas o estructuras agregando migraciones.
* **Artisan:** Línea de comandos de Laravel, empleado para ejecutar tareas y aplicaciones en laravel, tales como las migraciones. Ej: “php artisan migrate” esto permitiría ejecutar una migración para modificar, eliminar o agregar algo a la estructura de la base de datos.

Laravel también nos permite desplegar la aplicación en fase de desarrollo con un solo comando: **php artisan serve –-host=localhost –-port=80**. Con esta instrucción ya sería posible usar la aplicación desde el navegador web, esto evita tener que usar aplicaciones de terceros para ejecutar un servidor PHP.

### Bootstrap (CSS)

Según la página oficial, “Bootstrap es un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y JS”. Originalmente creado por y para Tweeter ®.

Es un framework para front-end que está desarrollado para dar soporte a la creación de sitios web dinámicos y aplicaciones web. Es unos de los preferidos por muchos por su facilidad para aplicarlo a casi cualquier entorno web.

Un modo de usarlo incluye archivos precompilados que permiten un uso rápido en cualquier proyecto web. Además de los archivos CSS y JS compilados y minificados. El otro modo de usarlo es usando el código fuente.

Su facilidad de uso permite combinar estilos que enriquecen el entorno visual de una página con muy pocos recursos.

## LA BASE DE DATOS

La aplicación creada hace uso de una base de datos relacional para el almacenamiento de los datos. Consta de 8 tablas interrelacionadas: *user, employee, absences, positions, roles, task\_positions, tasks, employee\_tasks*.Ver *Anexo 1*.

Para esta aplicación web se eligió MySQL como el lenguaje a implementar en la base de datos. Las principales razones para ello son:

* Velocidad
* Facilidad de uso
* Escalabilidad
* Disponibilidad (software libre)
* Fiabilidad
* Conectividad

Todas las tablas tienen una clave primaria de tipo entero autoincrementable. Algunas tablas tienen claves ajenas.

La estructura de la base de datos, tablas, relaciones y claves ajenas, son creadas usando el sistema de migraciones de Laravel. El *Anexo 2* es un ejemplo del código usado para crear la estructura de la tabla “tasks”

# ANÁLISIS DE REQUISITOS

## PLANTEAMIENTO DE NECESIDADES

Se requiere una aplicación para asignar tareas (también llamados servicios) a un grupo de personas. Todos los usuarios deben poder acceder con usuario y contraseña. Los usuarios básicos no pueden registrarse por su cuenta, el administrador se encargará de ello.

Para el adecuado funcionamiento de la aplicación hay que atender ciertos criterios al momento de la asignación. Entre las necesidades para asignar las tareas se establece que todas las personas tienen un Empleo (rango o categoría) y una antigüedad en el empleo.

Se debe llevar un conteo de cada tarea asignada a cada persona para que todos hagan la misma cantidad de tareas por cada tarea asignada.

Se debe poder incluir como ausentes a las personas que lo estarán por cualquiera que sea la razón en una fecha o rango de fechas. Si al momento de añadir a una persona esa persona ya tiene una tarea asignada, esa tarea debe anularse y se reasignarán todas las tareas ya asignadas a partir de esa fecha para conservar el conteo de cada tarea asignada.

Las condiciones para la asignación son las siguientes:

* + Si una persona estará ausente en una fecha específica no se le podrá asignar una tarea.
  + Si una persona estará asignada a una tarea cualquiera, ese día ya no se le podrán asignar más tareas.
  + Si una persona estará asignada a un servicio de armas, ese día y el día siguiente no se le podrán asignar más tareas.
  + Servicio de armas, 24 horas de duración:
    - El orden empieza por la persona con más antigüedad en el empleo.
    - Al terminar el servicio la persona tiene el resto del día libre y debe presentarse a trabajar al día siguiente.
  + Servicio de orden, termina antes de que acabe la jornada laboral:
    - El orden empieza por la persona más moderna en el empleo.
    - Al terminar el servicio la se incorpora a su puesto de trabajo y continúa con la jornada laboral si procede.

## QUIÉN PODRÁ HACER USO DE LA APLICACIÓN. CASOS DE USO

Se determina que podrá ser usada por dos usuarios. Uno con acceso básico y otro con privilegios de administrador que podrá acceder a todas las funciones de la aplicación.

**Usuario**: Puede ver sus propias tareas, ordenadas cronológicamente en orden descendente, un listado que muestra arriba las tareas próximas por hacer; y las tareas de todos los usuarios con tareas asignadas pendientes, ya que algunas tareas se deben realizar en grupo y es necesario ver el equipo de trabajo.

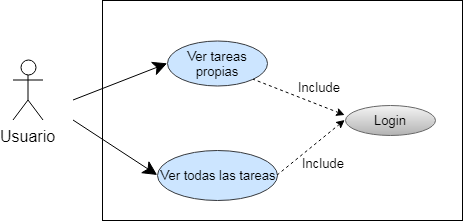


Ilustración 1, Casos de uso del usuario.

**Administrador:** Además de tener los mismos privilegios de un usuario normal, también puede: asignar, mostrar y borrar de tareas a los usuarios; CRUD de Tareas, Empleos, Relación tarea-empleo; CRUD de personal; alta, mostrar y borrar ausencias

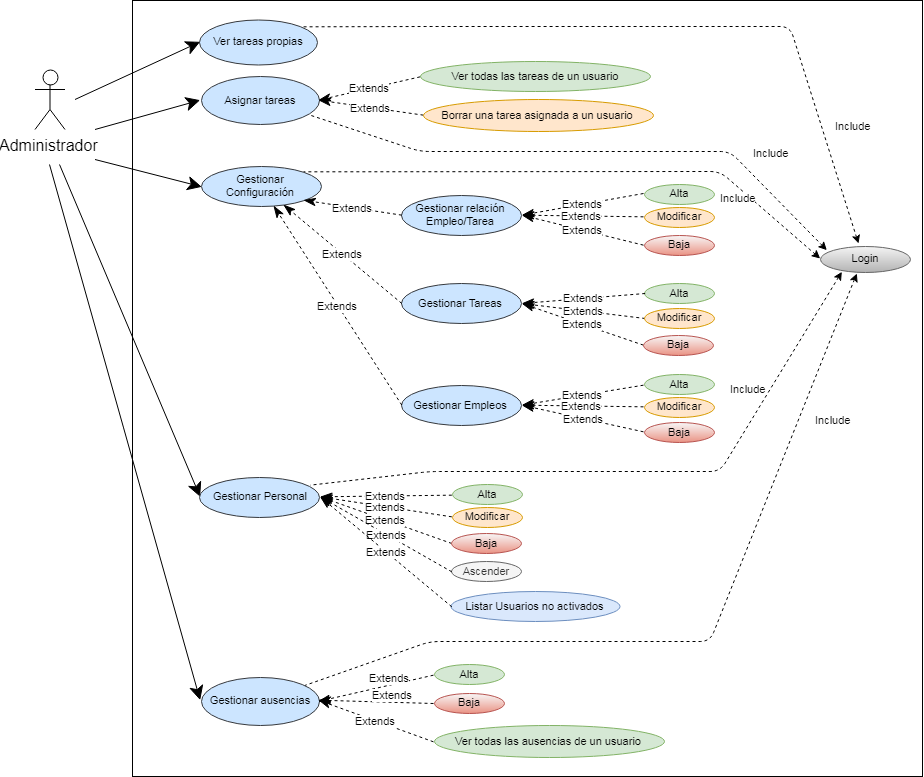
****

Ilustración 2, Casos de uso del administrador

# LA APLICACIÓN WEB

DESCRIPCIÓN

Esta aplicación web ha sido desarrollada para que un administrador asigne tareas a usuarios teniendo en cuenta su empleo (categoría), antigüedad en el empleo, la fecha en la que se realizará la tarea, y el número de veces que un usuario ha realizado una tarea.

CONSTITUCIÓN

Consta de una página inicial para que el usuario ingrese sus credenciales y pueda hacer uso de ella.



Ilustración , Página de Login

Una vez haya ingresado la página está dividida en Encabezado, Menú lateral y Cuerpo. Hay un menú en la parte izquierda que muestra 2 ítems si es un usuario, o el menú completo si es el administrador.

Únicas opciones para el Usuario:



*Ilustración 4, Menú de usuario, pantalla inicial*

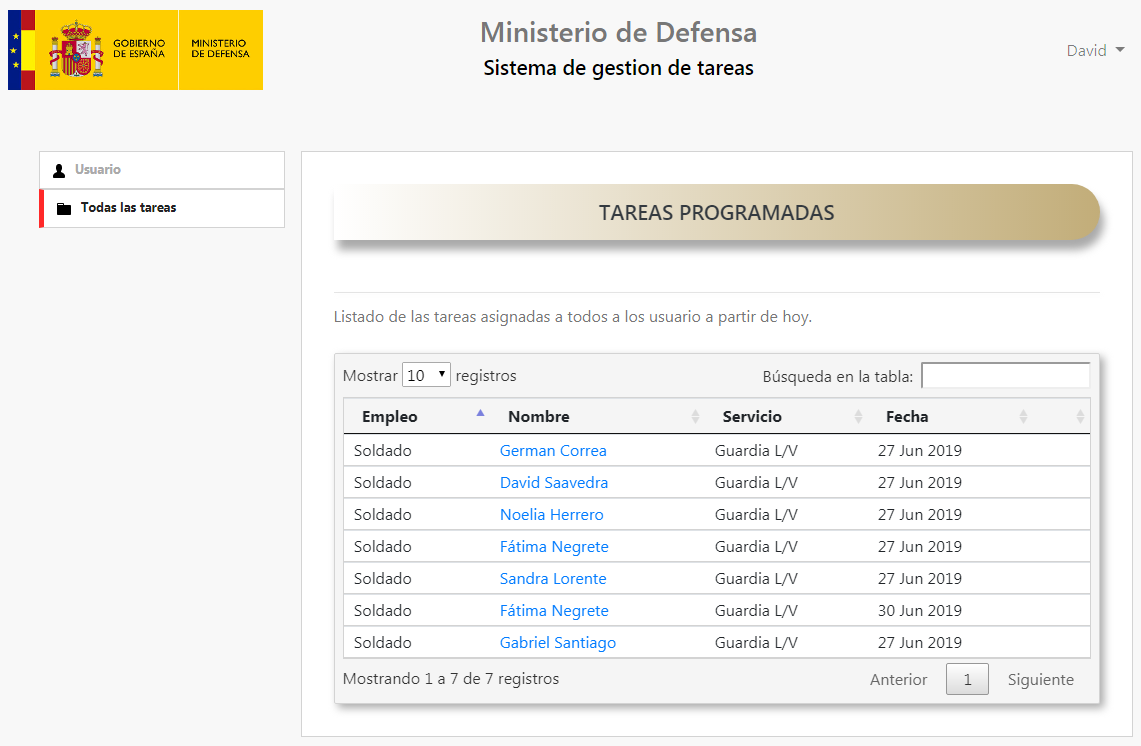


Ilustración , Todas las tareas

**Administrador**, al igual que al usuario al administrador también le aparecen tareas si las tuviese asignadas.



Ilustración , Menú administrador, pantalla inicial

PLANTILLAS DE VISUALIZACIÓN

Gracias a que se usa Blade como plantilla de visualización, la pantalla principal usa una única plantilla base que a medida que se llaman otras páginas conforman un todo, es decir, el Encabezado y el menú lateral izquierdo son una plantilla, cuando se presiona un botón del menú este carga una página que es un segmento que complementa la plantilla base.

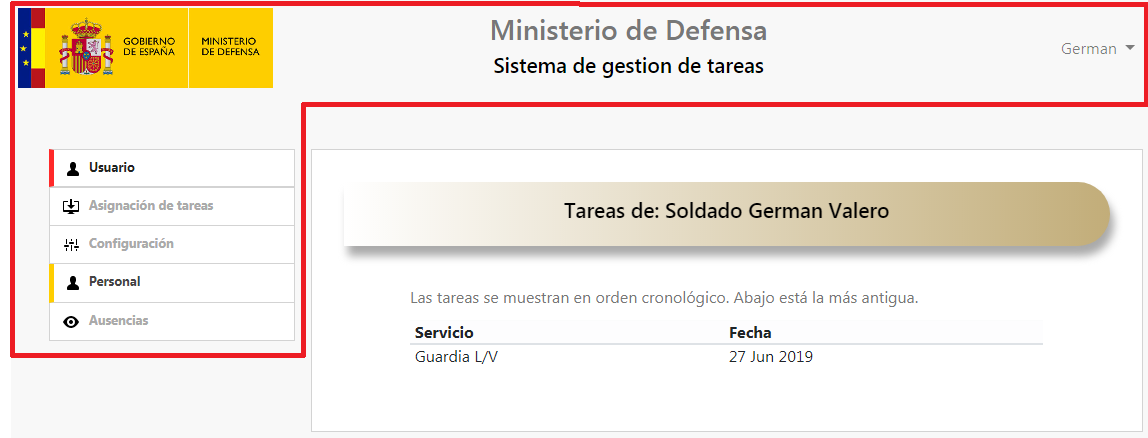


Ilustración 7, Plantilla base

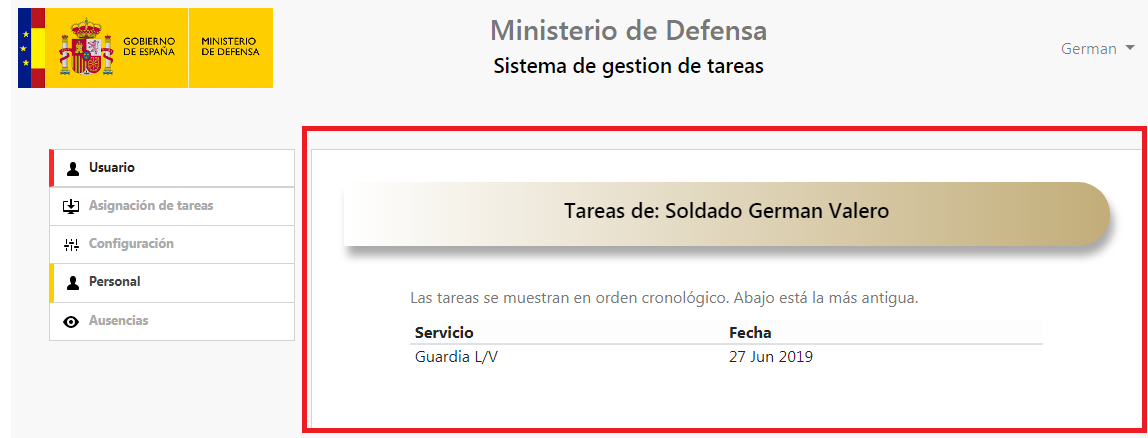


Ilustración 8, Complemento plantilla base



Ilustración 9, Complemento 2 plantilla base

ASIGNACIÓN DE TAREAS

Al momento de **asignar una tarea** se hace teniendo en cuenta los criterios anteriormente descritos en el PLANTEAMIENTO DE NECESIDADES. Los pasos son los siguientes:

* El administrador:
  + inserta la Fecha, Tarea, Empleo (se muestran sólo los que corresponden a la tarea, cargados a través de Ajax), y PAX.

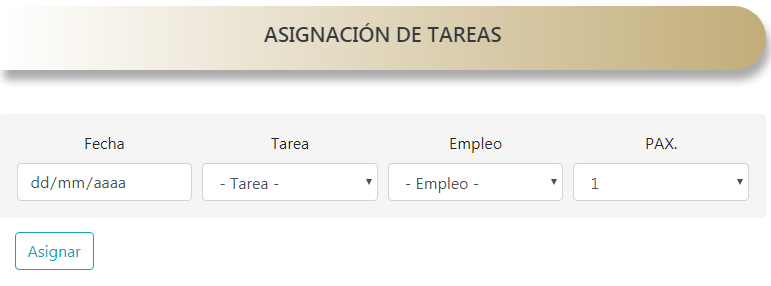


Ilustración 10, Asignación de tareas

* El sistema internamente
  + Hace listado: de los ausentes, de aquellos con tareas asignadas, de aquellos que tienen servicio de armas el día anterior, y los agrupa todos.
  + Compara el listado de todo el personal de ese empleo en la base de datos y descarta el listado anterior (todos los ausentes durante esa fecha), dejando únicamente el personal disponible y presente.
  + Ordena el listado del personal disponible según la tarea a realizar.
  + Obtiene el usuario y el número de veces que dicho usuario ha hecho la tarea que en cuestión. Visto gráficamente:

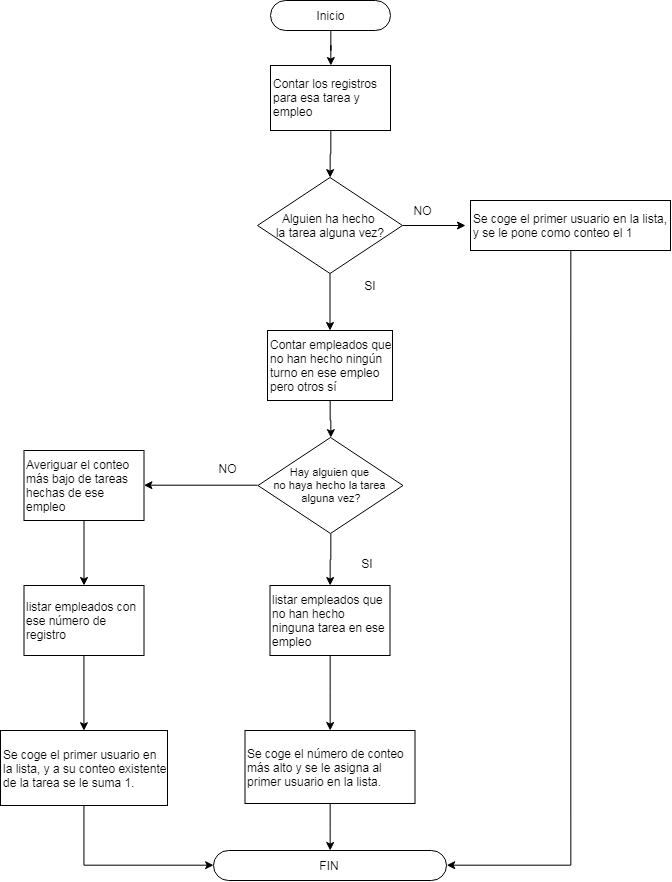


Ilustración , Diagrama para obtener número de conteo de tarea

* + Se almacena en la base de datos el usuario con el número de orden de la tarea.
  + Se muestra la pantalla donde se visualizan todas las tareas a partir de hoy.



Ilustración , Tareas asignadas

CONFIGURACIÓN

En esta sección se agregan, listan, actualizan y eliminan las tareas, empleos y la relación empleo-tarea. Esta sección debe manipularse con cuidado ya que la eliminación de un elemento de la base de datos pod

PERSONAL

AUSENCIAS

# CONCLUSIONES.

# TRABAJOS FUTUROS.

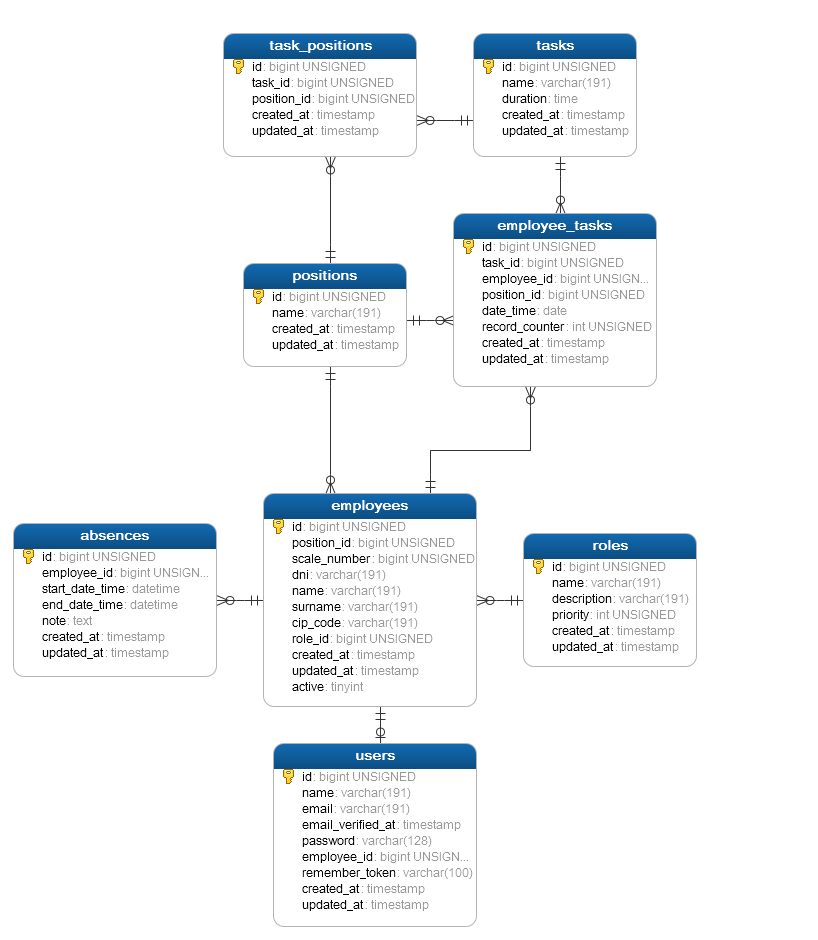
* Incluir una tabla de personas exentas de una tarea específica por condiciones extraordinarias durante un periodo determinado.
* Crear más roles intermedios dependiendo de las funciones.
* Escoger a un usuario específicamente al momento de asignar una tarea, podría ser un caso de voluntariado excepcional.
* Cambiar el idioma de la interfaz de usuario según corresponda a la configuración regional.
* El usuario podría hacer simulaciones de las posibles tareas en un futuro para prever cuándo se vería afectado.
* El usuario solicita o comunica su ausencia en una fecha específica.

# BIBLIOGRAFÍA.

* <https://techterms.com/definition/javascript>
* <https://whatis.techtarget.com/definition/PHP-Hypertext-Preprocessor>
* <https://www.techopedia.com/definition/3929/javascript-js>
* <https://www.tutorialspoint.com/ajax/what_is_ajax.htm>
* <https://www.w3schools.com/xml/ajax_intro.asp>
* <https://developer.mozilla.org/es/docs/Glossary/jQuery>
* <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>
* <https://codigofacilito.com/articulos/que-es-html>
* <https://www.w3schools.com/html/html_intro.asp>
* <https://www.esepestudio.com/noticias/que-es-mysql>
* <https://www.larashout.com/what-is-laravel-and-why-you-should-learn-it>
* <https://getbootstrap.com/>
* <https://vmokshagroup.com/blog/bootstrap-advantages/>

# ANEXOS

### Anexo 1 – Relaciones de la base de datos.



### Anexo 2 - Código para crear la tabla “tasks” usando la migración de Laravel

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateTasksTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('tasks', function (Blueprint $table) {

$table->bigIncrements('id');

$table->string('name');

$table->time('duration')->nullable();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('tasks');

}

}

# MANUAL INSTALACIÓN/CONFIGURACIÓN/USO.

 Partes de código/Ficheros de configuración importantes.