



**Autores**: Gloria Taboada, Carlos Gómez y Oscar Robles

**Ciclo**: Grado Superior de Desarrollo de Aplicaciones Web (2º curso)

**Clase**: MP12 - Proyecto

**Tutor/a**: Francesca Tomás Artigues

INDICE

1. Introducción.

1.1. Propósito del proyecto.

Este proyecto nace de la idea de comodidad a la hora de reservar una habitación on-line, ampliando dichas reservas a todos los servicios ofrecidos por el hotel-balneario, tanto de servicios de habitación como de tratamientos de belleza, pasando por reservas de mesa en sus restaurantes o actividades que realice.

La idea es que el cliente del balneario pueda reservar todos los servicios on-line, escogiendo el horario disponible que más le convenga, pudiendo cancelarlo en cualquier momento sin necesidad de llamar por teléfono.

1.2. Objetivos.

El objetivo del proyecto es facilitar la gestión del personal encargado de hacer las reservas, ahorrando empleados y tiempo al delegar dicha función en el cliente.

El hotel-balneario pondrá a disposición de sus clientes una agenda encargada de mostrar las fechas y horarios disponibles para sus habitaciones, restaurantes y/o tratamientos ofrecidos.

Los clientes tendrán que registrarse en la aplicación para poder gestionar sus reservas.

Como política de reservas y a fin de que el cliente no pueda acaparar la agenda, se penalizará con el pago de un porcentaje del importe del servicio contratado por reserva anulada sin suficiente anterioridad o según las normas del hotel.

El cliente que no esté registrado y no desee registrarse no podrá hacer reservas. De esta manera incentivamos a los clientes a registrarse y a facilitarnos sus datos.

2. Desarrollo.

2.1. Recogida de requisitos: alcance, requisitos de datos y requisitos funcionales.

Interfaz para el cliente

El proyecto cuenta con una interfaz usable, ágil y de fácil navegación para el cliente. Las diferentes imágenes de cabecera y colores utilizados facilitan la vista del apartado donde se encuentra el cliente.

El menú de la barra de navegación siempre esta visible en cada apartado para una rápida navegación, al igual que el acceso de usuarios registrados o el cambio de idioma.

El diseño de cada apartado está cuidadosamente diseñado para un mayor atractivo a la vista del cliente, usando colores relajantes y diferentes tipos de fuente proporcionadas por *Google Fonts*, que están siempre disponibles para cualquier dispositivo con el que se navegue.

El diseño de la aplicación es totalmente *responsive*, atendiendo a la política *First Mobile.* Las imágenes y las fuentes utilizadas están en medidas escalables, como porcentajes o *vw (ancho del viewport),* que permiten ser expuestas en diferentes tamaños sin pérdida de calidad en su representación.

Toda la aplicación está basada en bootstrap 4.

El uso de ventanas modales para mostrar el detalle del producto seleccionado con sus imágenes correspondientes agiliza la carga de cada vista de la aplicación.

La tarea de reserva está simplificada al máximo con sus respectivas notificaciones.

El área personal del cliente registrado está diseñada para que pueda modificar sus datos personales y obtenga un recordatorio de sus anteriores reservas.

Interfaz para el administrador

El proyecto cuenta con un menú personalizado para el administrador de la aplicación que le permite ejecutar tareas de administración como la gestión de empleados, clientes y horarios.

Actores y guiones de la aplicación

|  |  |
| --- | --- |
| Actor | Administrador |
| Descripción | Usuario externo que interactúa con la aplicación para administrarla. |
| Guion | * Tiene control total sobre toda la aplicación. * Puede crear, modificar y eliminar cualquier usuario. * Asigna permisos a todos los usuarios. * Puede gestionar reservas. * Puede modificar horarios. |

|  |  |
| --- | --- |
| Actor | Usuario |
| Descripción | Usuario externo que interactúa con la aplicación. |
| Guion | * Puede mirar toda la información de la aplicación que no requiera permisos de administrador. * Puede registrarse en la aplicación. * Puede modificar sus datos de perfil. * Puede hacer y gestionar sus reservas. * Puede anular reservas. |

|  |  |
| --- | --- |
| Actor | Visitante |
| Descripción | Usuario externo que interactúa con la aplicación. |
| Guion | * Puede mirar toda la información de la aplicación que no requiera permisos adicionales. |

Objetivos técnicos alcanzados, no alcanzados, ampliados.

Objetivos técnicos alcanzados

Los objetivos alcanzados son haber empezado un proyecto desde cero, planificando cada apartado y basándonos en un diseño creado a partir de una imagen adjuntada en la propuesta de proyecto, clonándola en formato web con la ayuda de css y bootstrap 4, herramienta que hemos aprendido a usar por nuestra cuenta con espectaculares resultados, ya que durante el curso hemos utilizado la versión 3.

El sistema de autentificación diferencia entre logueo de usuarios y logueo de administrador, teniendo este último todas las herramientas para crear, editar y eliminar usuarios en ventanas modales bajo petición Ajax y así optimizar la carga de información sin requerir muchos recursos.

La aplicación está diseñada para abarcar varios idiomas.

\*\*\*\* AQUÍ PODEIS PONER ALGO MAS \*\*\*\*\*

Objetivos técnicos no alcanzados

Como objetivo técnico no alcanzado sería la implementación de un calendario donde se reflejaran los días que ya están ocupados.

No obstante es un objetivo no alcanzado por falta de tiempo y no por falta de conocimientos.

Objetivos técnicos ampliados

Un objetivo técnico ampliado a tener en cuenta es el sistema empleado para traducir el contenido de la página, que facilita mucho la tarea, pudiendo traducir a cualquier idioma el contenido de un apartado solo con añadir un nuevo archivo con la traducción del texto deseado.

Posibles ampliaciones futuras

Una posible ampliación futura sería añadir los idiomas más usados habitualmente.

Otra posible ampliación sería enviar un email al usuario que quisiera reservar habitación u otro servicio no habiendo estado esta disponible para las fechas seleccionadas, que previamente habría cancelado otro usuario.

Valoración personal Gloria Taboada

Mi valoración personal es muy positiva, teniendo en cuenta que este proyecto cuenta con un gran peso en la parte de diseño, cosa que me encanta desarrollar.

Creo que una cosa fundamental a la hora de crear una aplicación es hacerla atractiva a la vista, facilitando así su navegación por los diferentes apartados.

Desde el principio hemos coincidido todos los compañeros en las herramientas con las que trabajaríamos para crear el proyecto, que son las aprendidas durante el curso, y cada uno es más habilidoso en unas o en otras, así que el reparto de tareas ha sido fácil.

PARA EL POWERPOINT (parte que yo explicaré)

1 diapositiva:

* Diseño atractivo
* First Mobile (responsive)
* Bootstrap 4

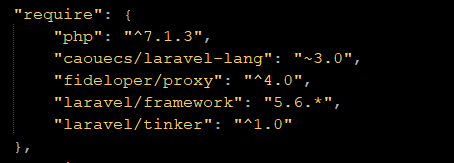
2 diapositiva:

* Ventanas modales
* AJAX

Instalación del paquete notificaciones.

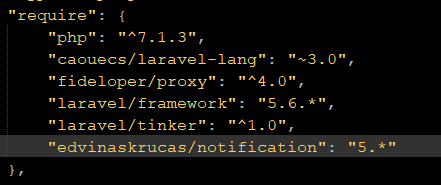
Para instalar el paquete "Notification" de edvinaskrucastendríamos que editar

el fichero composer.json y en su sección require añadir la siguiente línea:





Al añadir el paquete al fichero composer.json tenemos que llevar mucho cuidado de añadir una coma como separador de los elementos de la sección require, **exceptuando** el último elemento de la lista, que no tendrá que llevar coma al final (sino nos dará error).



Una vez añadido ejecutamos el siguiente comando en la carpeta donde está nuestro proyecto para instalar el paquete:

composer update



Y después de instalar tendríamos que editar el fichero config/app.php para añadir, en la sección providers , la siguiente línea:

Krucas\Notification\NotificationServiceProvider::class,

Y en la sección aliases lo siguiente:

'Notification' => Krucas\Notification\Facades\Notification::class,

Además también tenemos que añadir el middlewareal fichero app/Http/Kernel.php . En el array de $middlewareGroups añadimos la siguiente línea al grupo web , después de 'Illuminate\Session\Middleware\StartSession':

\Krucas\Notification\Middleware\NotificationMiddleware::class,

Con esto ya tendríamos instalada esta librería y podríamos empezar a utilizarla. Por ejemplo, para añadir una notificación desde un controlador podemos utilizar los siguientes métodos:

Notification::success('Success message');

Notification::error('Error message');

Notification::info('Info message');

Notification::warning('Warning message');

Importante: tenemos que acordarnos de añadir el espacio de nombres de esta clase ( use Notification; ) en el controlador donde la vayamos a utilizar, de lo contrario nos aparecería un error.

Y para mostrar las notificaciones, desde una vista añadiríamos el siguiente código:

{!! Notification::showAll() !!}

O también podemos usar la extensión que incorpora para Blade y simplemente añadir:

@notification()