

**NotApp**

Despliegue de Servicios Multimedia



Universidad pública de navarra

María Andrés

Alicia Fernández

Amaia Iturri

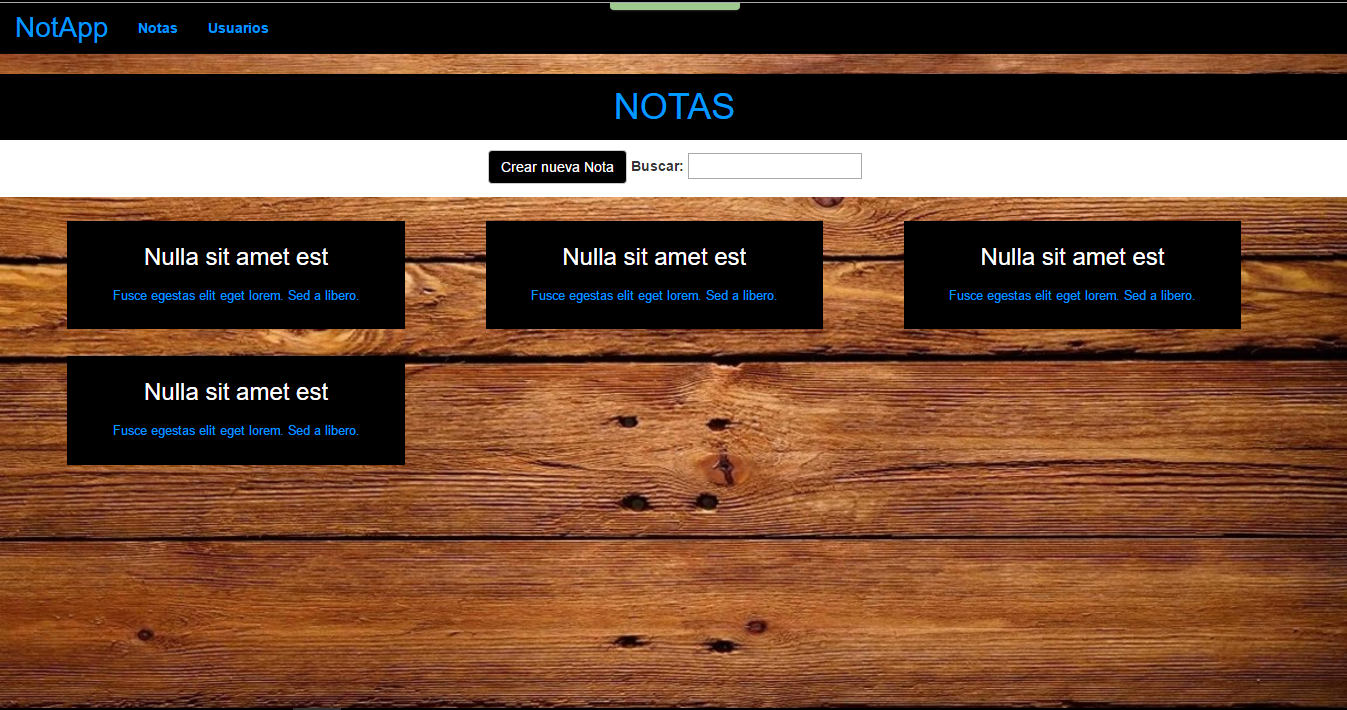


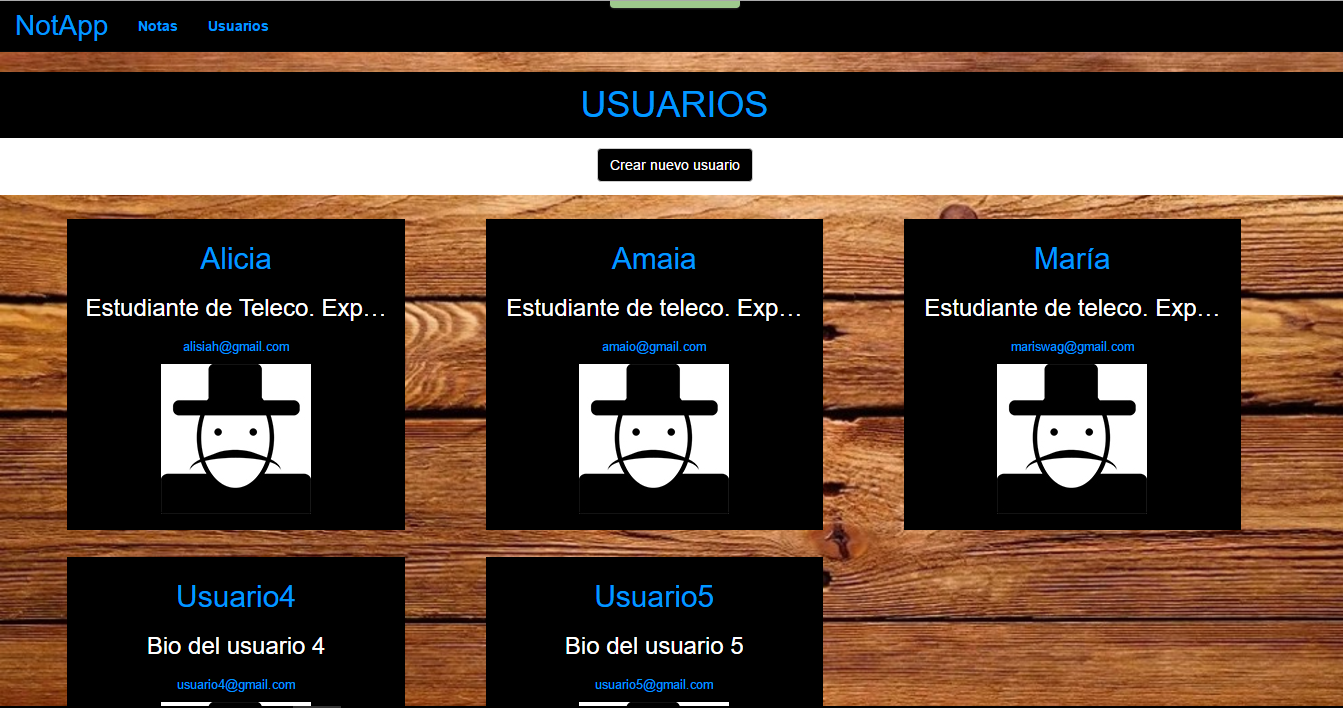
# Introducción

El objetivo de este trabajo es desarrollar un ‘Task Manager’ con una serie de requisitos como mostrar un listado de tareas y otro de usuarios, poder crear editar y borrar en ambos listados, utilizando todos los conceptos teóricos vistos en la asignatura y por supuesto, realizando un diseño responsivo que se adapte a todas las pantallas.

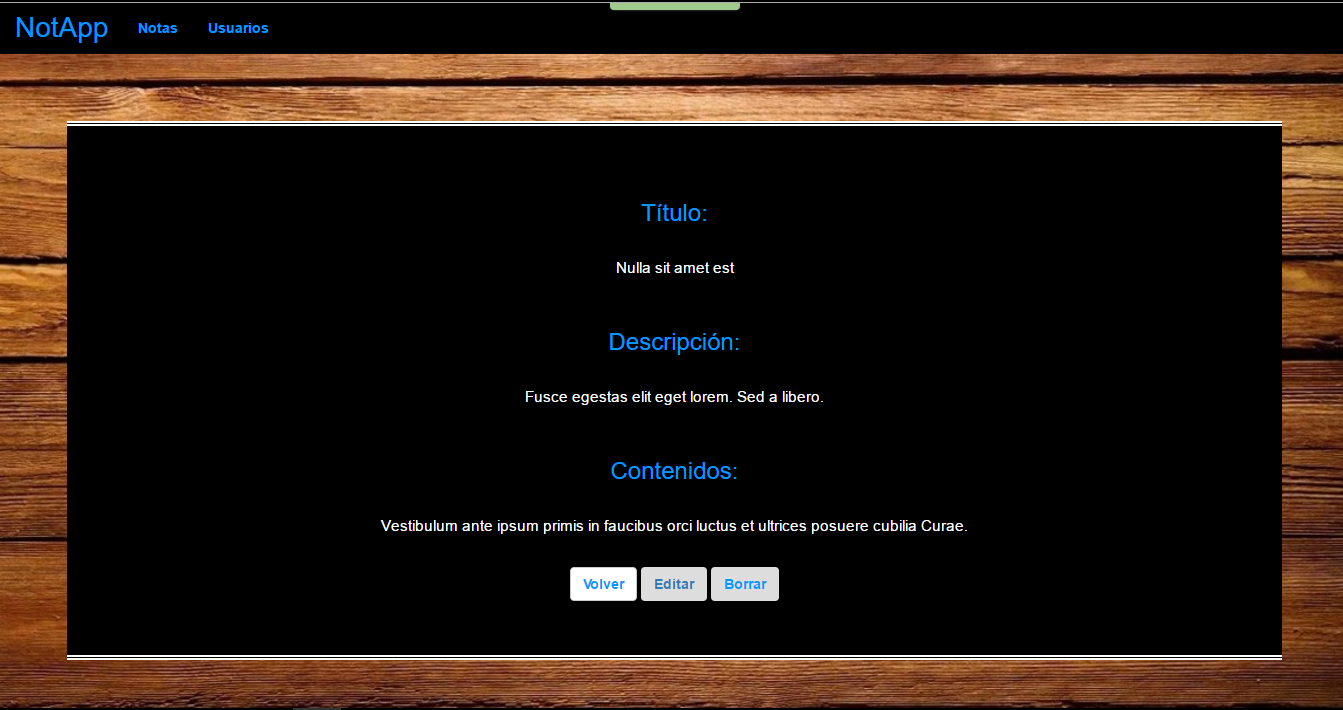
Presentamos **NotApp** como nuestro ‘Task Manager’. En el apartado de usuarios se puede crear, borrar y actualizar. En el apartado de notas se puede además buscar.

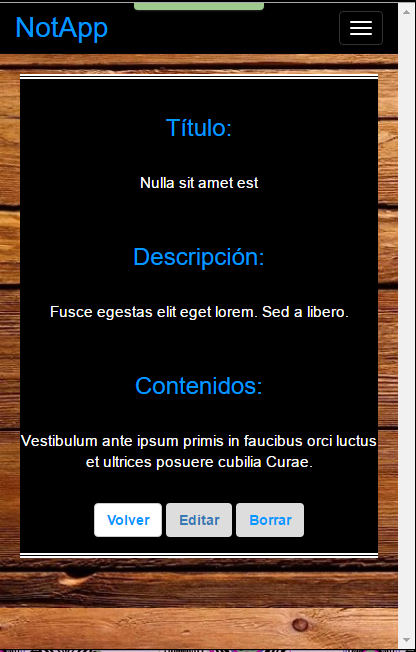
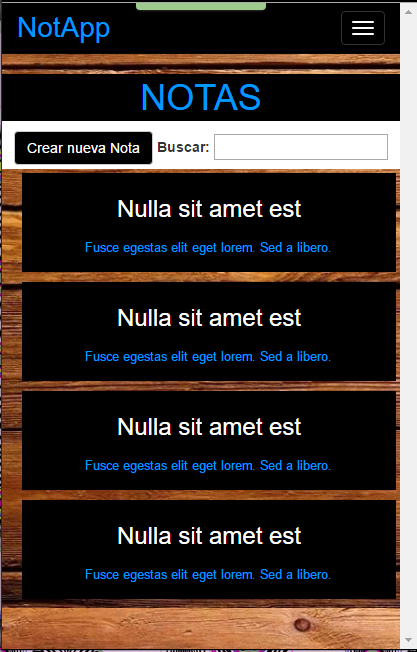
La interfaz de usuario trata de ser sencilla, clara y elegante como se muestra a continuación:





En cuanto a la interfaz gráfica se le han añadido una serie de detalles como, por ejemplo, al pasar el ratón por encima de un usuario o nota aparece un borde azul sobre la nota/usuario que está siendo señalado. Además, cuando un nombre o descripción corta es demasiado grande, se cortará añadiendo unos puntos suspensivos. Se podrá ver al completo clicando en la nota/usuario donde nos aparecerá toda la información sobre la nota o usuario seleccionado.



Al cambiar la resolución de la pantalla (haciéndola más pequeña) en vez de aparecer las pestañas de “Notas” y “Usuarios” aparece un botón con tres barras horizontales a la derecha de la pantalla. Al clicar se despliegan las pestañas que se acaban de comentar para poder navegar por ellas.

# Desarrollo

Se cuenta con cuatro controladores diferentes para cada parte, uno para el índice, otro para editar, otro para crear una nueva y otro para mostrar una única nota. En todos ellos se utiliza $location.path("/"); para volver al índice o al apartado correspondiente en cada caso.

* En notas-index-controller.js (y el correspondiente para usuarios), mediante el método query, llamamos al servidor a ‘app.get('/notas)’. Desde el servidor llamamos a la base de datos de mongo y recogemos en un json todos los datos. Esos datos serán mostrados al inicio, cada uno en su apartado (notas o usuarios).
* En nota-nueva-controller.js (y el correspondiente para usuarios), mediante el método $save, se mandan los datos al servidor. Para ello se ha creado la función guardaNuevaNota, a la cual se accede mediante el botón de Crear nota en el html correspondiente. El servidor guardará los datos en mongo creando una nueva nota.
* En nota-editar-controller.js (y el correspondiente para usuarios), se realizan dos cosas, coger los datos de la nota en función del id para mostrarlos en el formulario de actualizar y una vez modificados los campos deseados al presionar el botón Actualizar llamará a la función actualizaNota la cuál utiliza el método update para llamar al servidor a app.put('/notas'). Una vez en el servidor se busca por id la nota en mongo y se modifican y guardan sus parámetros.
* En nota-show-controller.js (y el correspondiente para usuarios) se cogen los datos de una nota en concreto en función del id. Dentro de este controlador es donde se realiza también la función de borrarNota, la cual eliminará la nota en caso de ser presionado el botón de borrar. Para borrar se utiliza el método delete, el cual llama al servidor a app.delete('/notas/:id'), donde se borra la nota de la base de datos.

Tanto para notas como para usuarios hay un archivo javascript de servicios donde se crea un Factory, para indicar los servicios que tiene que seguir los diferentes métodos antes mencionados.

En el archivo routes.js se indican las rutas necesarias para esta aplicación, dentro de la ruta se indica el archivo html que debe usar y el controlador. En caso de que se escriba mal una ruta se manda por defecto a la página principal de notas.

Se utilizan dos directivas, una para usuarios y otra para notas. Dentro de las directivas se indican los parámetros que se usan. Se usan el id, titulo, contenido y descripción para notas y para usuarios el id, nombre, biografía y email.

Al crear un nuevo usuario o nota se valida que se introduzca un título (o nombre), ya que sin él no hay nota o usuario.