

Universidad Nacional de General Sarmiento

Tecnicatura Universitaria en Informática

Introducción a la Programación

Comisión: 14

Docentes: Bidart Gauna Lucas Ezequiel y Mieres José Luis

Banda Horaria: lunes de 13hs a 17hs

Estudiantes: Zablosky Alan, Machado Hernán y Aldeco Becette Joaquín

**Mails: alan_zablosky@hotmail.com, hernan_onlyarg@hotmail.com,
joaquinaldeco@gmail.com**

Trabajo Práctico Integrador:

Fecha de Entrega: 24 de junio de 2024



**Universidad
Nacional de
General
Sarmiento**

En este informe se hará una introducción acerca del código implementado en la realización del trabajo práctico. Se explicarán las funcionalidades de lo que ha sido agregado y las dificultades que tuvimos durante la realización del trabajo mencionado.

El trabajo consiste en hacer una página web que busca y filtra fotos de la nasa. En un principio ya fue hecha la petición y la implementación de código estructural de la página. Nuestro trabajo consistió en saber qué código hay que reutilizar, cuando y de qué manera hacerlo sabiendo que: lo pedido es que se muestren las imágenes, que se puedan filtrar con un buscador, que se creen usuarios y tengan su apartado de favoritos, entre otras cosas adicionales.

Nosotros pudimos realizar lo solicitado y adicionalmente le agregamos un traductor al buscador, el envío de credenciales por mail y una breve modificación del diseño de página.

Explicación del código(brevemente):

Primero las funciones de **getAllImages**, **getAllImagesAndFavouriteList** y **home** lo que hacen es traer las imágenes desde el api de la nasa y mostrarlas.

Para hacer el **login** (el inicio de sesión) ya estaba casi todo hecho, nosotros solo tuvimos que importar una función desde las urls del main y llevarla a las urls de nasa gallery. Además, tuvimos que agregar el redireccionamiento de la url de login desde el header.html. Importante ver en settings a dónde se redirecciona cuando uno llena el formulario de ingreso. Para el **logout** fue algo similar, aunque para hacer funcionar la función **exit** tuvimos que sacarle el requerimiento de que el usuario esté logueado para entrar. Creemos que la función exit podría no ser necesaria, pero la hicimos funcionar porque era de las que ya estaban escritas.

Luego la función **search**, básicamente lo que hace es buscar lo que vos escribas en el buscador, pero hay que comprender que, si el “cerebro de la nasa” no reconoce la palabra que estas buscando, entonces no lo puede filtrar. Para ello lo que aplicamos es una lista con muchas palabras (se encuentra en el archivo **palBuscables.py**) que ya corroboramos que sí traen imágenes como resultado de la búsqueda, de esa manera funciona como un “embudo”, ya que no sabemos con exactitud cuales palabras sí busca el “cerebro de la nasa”, lo mejor es darle las palabras nosotros y decirle que si no son esas palabras, devuelva una búsqueda por defecto.

Adicionalmente a la función search le agregamos un **traductor de Google Translate** que si buscas las palabras en español te las traduce al inglés antes de enviar la petición.

Siguiendo tenemos las **funciones de favoritos** (**getAllFavouritesByUser**, **saveFavourite** y **deleteFavourite**). Estas funciones se sostienen de mucho código anterior que se encuentran en repositories, models, mapper y service nasa. Nosotros en views solo hacemos las invocaciones, pero es en **service**

nasa donde realmente hacemos lo que se necesita. Para **saveFavourite** lo que hacemos es guardar la petición asociándola al usuario (se guarda en la base de datos). Luego para **getAllFavouritesByUser** lo que hacemos es decirle que, si el usuario está autenticado, filtre de entre la base de datos todos los favoritos que estén asociados al usuario que hace la petición. Esos favoritos se los recorre y se los guarda en forma de diccionario para luego ser transformados en nasa card, que luego se mostrarán en pantalla. Por último, la función **deleteFavourite** lo que hace es buscar a quien está asociado el favorito en cuestión y los desvincula (lo borra de la base de datos de favoritos al favorito).

Por último, la función **register_view** tiene dos funcionalidades, el registro de nuevos usuarios y el **envío de credenciales por mail**. Para la parte del registro creamos un register.html con un formulario de registro, este formulario toma los campos a rellenar desde models, ya que implementamos el código necesario para la creación de un nuevo usuario. Luego lo que hace la función register_view es que, si se envía el formulario y este mismo es válido, entonces lo guarda en la base de datos para que ya pueda iniciar sesión. Adicionalmente, tomamos del formulario el apartado “email” y le enviamos a su mail sus credenciales (el usuario y contraseña).

Dificultades durante la realización del trabajo:

El trabajo nos resultó difícil, entre las distintas dificultades que cada uno tuvo, hay algunas en las que coincidimos. En primer lugar, el nivel del trabajo práctico era muy superior al de la cursada, no solo a nivel lógica sino también en lenguajes, frameworks, etc. Fue mucho lo que tuvimos que buscar por nuestra cuenta, creemos que la cursada nos prepara muy poco como para ser capaces de realizar este trabajo.

Luego se puede mencionar como un obstáculo en sí la descarga del trabajo práctico, fueron necesarios unos cuantos intentos hasta lograr que se muestre desde donde ya se puede arrancar a editar código.

Otro aspecto a mencionar como dificultad es saber como arrancar cada función, o cual es el “camino correcto” ya que para comprobar si algo funcionaba había que completarlo en su totalidad y que esté perfecto para poderlo “correr” al programa y que ande, en el mientras tanto saltan muchos errores y cuesta saber si vas bien en el camino en el que vas o si tendrías que borrar y empezar de nuevo.

El aprender a usar los repositorios locales y remotos es en sí uno de los puntos más difíciles del trabajo.

Por último, cuando ya la cabeza no te da más de intentar e intentar y el programa no responde, tenemos que recurrir a fuentes externas (no siempre en español) para intentar dar con la solución (o llevarte un virus de por medio).