

Introducción al Desarrollo Web/Móvil

Taller III

Incorporando servicios externos

MobileHub

Descripción:

La gestión del código fuente (SCM) es una parte fundamental del desarrollo de software y aplicaciones en general, exponentes como Git han cambiado la forma de abordar esta necesidad mediante un sistema de control de versionado distribuido (DVSC), a su vez, plataformas como GitHub han llevado el concepto distribuido a un paso superior aprovechando las capacidades de la web y el robusto diseño de Git.



Logo MobileHub

Dado el auge de aplicaciones móviles, se ha solicitado a los estudiantes de **Introducción al Desarrollo Web/Móvil** de la Universidad Católica del Norte diseñar una solución que permita visualizar todos los commits realizados por un usuario, en un repositorio específico que se encuentre en la plataforma github.com.

Se solicita a los/as estudiantes trabajar en una aplicación **móvil**, para ello deber armar una **API RESTful** que permita la comunicación entre el **Frontend móvil** y el **Backend**. **Utilizar cualquier framework de backend y frontend.**

El sistema permitirá a los usuarios autenticados visualizar todos sus repositorios en una lista, al seleccionar alguno de ellos se deberá redirigir a una sección que contenga todos los commits del repositorio y su detalle.

Las funciones de solicitadas son las siguientes:

1. **Iniciar aplicación:** El sistema, al iniciar, deberá mostrar una página de inicio que contenga el nombre de la aplicación (MobileHub), el logo y 2 botones: Iniciar sesión y registrase.

- a. **Sobre los botones:** El botón de iniciar sesión debe ser primario, mientras que el de registrarse debe ser secundario ([jerarquía de botones](#)).
- 2. **Iniciar sesión:** El sistema solicita el correo electrónico de la UCN y, a modo de contraseña, el RUT sin puntos ni guion del usuario. Si las credenciales coinciden se retorna un JWT con su correo electrónico. En caso contrario deberá informar al usuario que las credenciales son inválidas. En esta vista deberá existir un enlace ([botón terciario](#), a veces llamado [links](#)) que permita recuperar contraseña en caso de olvido, esto siguiendo las reglas de la funcionalidad **Recuperar contraseña**.
- 3. **Registrarse:** El sistema permite a los usuarios registrarse con su correo electrónico si y solo si es dominio de la **UCN**, nombre completo, año de nacimiento y RUT. La contraseña será RUT sin puntos ni guion. Todos los errores en los campos deberán ser informados al usuario con mensajes específicos por cada campo.
 - a. **Sobre el correo:** El sistema, para validar, deberá enviar un enlace al correo electrónico ingresado el cual dirigirá al usuario a una página de confirmación para habilitar la cuenta. Este deberá tener una expiración de 5 minutos pasado el envío.
 - b. **Sobre el RUT:** Se deberá ingresar con puntos y guion, el sistema deberá calcular y revisar si el dígito verificador es válido.
 - c. **Sobre el año de nacimiento:** El sistema no permite años de nacimiento inferiores a 1900 ni superiores al año actual.
 - d. **Sobre el nombre completo:** El sistema no permitirá nombres con menos de 10 caracteres de longitud y con más de 150.
- 4. **Visualizar repositorios:** El sistema permite, a los usuarios autenticados, visualizar todos los repositorios del usuario que ha iniciado sesión, estos deberán estar listados por última fecha de modificación.

- a. **Información que desplegar:** Cada repositorio deberá mostrar su nombre, fecha de creación y cantidad de commits hasta la fecha.
- 5. **Visualizar commits de un repositorio:** El sistema permite, a los usuarios autenticados, visualizar todos los commits de un repositorio luego de haberlo seleccionado en el listado de repositorios. El listado de commits se muestran por orden de creación, donde el más reciente aparece más arriba.
- 6. **Editar información de perfil:** El sistema permite, a los usuarios autenticados, al usuario editar su nombre completo, correo electrónico y año de nacimiento. Todo esto siguiendo las validaciones de registro.
- 7. **Actualizar contraseña:** El sistema permite, a los usuarios autenticados, que actualicen su contraseña, el sistema ya conoce el correo del usuario por lo que no lo solicitará.
- 8. **Cerrar sesión:** En esta opción, el sistema debe permitir al administrador cerrar su sesión actual, volver a la pantalla de inicio de sesión y borrar el JWT. **Se debe validar que no pueda volver a ingresar a las funcionalidades de usuarios autenticados sin antes volver a iniciar sesión.**

Consideraciones:

- Para obtener la información relacionada a commits y repositorios se **debe** utilizar la [API](#) de GitHub.
- El registro y edición del perfil de usuario debe seguir las reglas:
 - El correo debe ser único en el sistema.
 - El RUT debe ser único en el sistema.
 - La validación de dígito verificador del RUT debe seguir el algoritmo del [módulo 11](#).
 - El correo debe ser válido ([expresión regular](#)) y perteneciente al dominio UCN definido a continuación: @ucn.cl, @alumnos.ucn.cl, @disc.ucn.cl, @ce.ucn.cl.
- El proyecto debe poseer una estructura inicial que incluya tanto Backend como Frontend móvil en una sola carpeta.
- Se debe usar Git para versionar su trabajo, debe ser solo un repositorio para ambos componentes del taller.
- Se debe desarrollar la interfaz de usuario utilizando el framework escogido para el Frontend, además se debe integrar un framework o librería de componentes para el diseño.
- La aplicación debe tener un diseño agradable para la vista de un usuario, debe tener un diseño elegante y vistoso.
- Se deberá utilizar JWT para la autenticación y autorización.
- Las contraseñas del sistema deben estar encriptadas utilizando BCrypt o algún formato similar.

Entregables:

La entrega del taller debe contener:

1. Ambos componentes, Backend y Frontend en un mismo repositorio de **GitHub**.
2. Archivo .json de **Postman** que haya utilizado para realizar pruebas del Backend.
3. Archivo **README.md** en la raíz del proyecto con las instrucciones para levantar los proyectos de Frontend y Backend. Estas instrucciones deben estar dirigidas a un usuario que **no tiene las tecnologías** ocupadas **instaladas** en su máquina personal.

Condiciones de entrega:

- La fecha de entrega del taller es a más tardar el viernes **15 de diciembre a las 23:59 hrs.**
 - No se aceptarán entregas fuera de plazo o cuyos “commit” de GitHub se realicen fuera de la fecha de entrega.
- La resolución del taller es de manera individual.
- Consultas sobre el taller serán respondidas vía Campus Virtual UCN y por correo electrónico a los ayudantes de la asignatura:
 - David Araya: david.araya@alumnos.ucn.cl
 - Marcelo Céspedes: marcelo.cespedes@alumnos.ucn.cl
- La entrega del taller será por medio de GitHub subiendo la URL del repositorio **privado** a la entrega en Campus Virtual e invitando a los ayudantes como colaboradores a su repositorio.
- La copia del taller será sancionada con nota 1,0 y los antecedentes del caso serán reportados a jefatura de carrera y a registro curricular.
- Si la aplicación no es posible ejecutarla siguiendo los pasos establecidos en el entregable del README.md, **no se procederá a calificar y se colocará la nota mínima 1,0.**

Penalizaciones:

- El no utilizar JWT implicará un **descuento de 20 décimas.**
- El no encriptar contraseñas implicará un **descuento de 20 décimas.**
- No documentar implicará un **descuento de 30 décimas.**
- No utilizar la [API](#) de GitHub implicará **nota 1,0.**
- No invitar al ayudante [David](#) al repositorio privado antes de la revisión tendrá un **descuento de 20 décimas.**
- No almacenar el secreto del JWT en algún archivo .env o similar (que lo mantenga seguro) implicará un **descuento de 30 décimas.**

Rúbrica de evaluación:

- La Tabla I corresponde a la rúbrica del taller, la nota del taller será calculada sobre el puntaje total con un 60% de exigencia.

Criterio	Excelente (7,0 - 6,0)	Aceptable (6,0 - 3,5)	Requiere Mejorar (3,5 - 1,0)
Interfaz Web	La interfaz web corresponde al menos al 85% de lo requerido por el taller.	La interfaz web corresponde entre un 60% y 85% de lo requerido por el taller.	La interfaz web corresponde a menos del 60% de lo requerido por el taller.
API	Se construye todos los EndPoint correspondientes para dar funcionalidad a los requerimientos del taller.	Falta al menos un EndPoint correspondiente para dar funcionalidad a los requerimientos del taller.	No se establecen los EndPoint para dar funcionamiento a lo requerido por el taller
Estructura del Taller	La estructura del proyecto contempla un proyecto independiente para el backend, un proyecto independiente para el frontend. Los dos proyectos se encuentran en una sola carpeta versionada en git.	La estructura del proyecto contempla un proyecto independiente para el backend, un proyecto independiente para el frontend. Los dos proyectos se encuentran en carpetas separadas versionada en git.	No se establece una estructura clara del proyecto. No se encuentra en un solo repositorio de git
Implementación de Funcionalidades	La aplicación cumple con todos los requisitos funcionales solicitados	La aplicación no cuenta con al menos 2 requisitos funcionales.	La aplicación no cuenta con más de 2 requisitos funcionales
Git	La aplicación web se encuentra versionada correctamente con GIT en GITHUB, Bitbucket o cualquier otro versionador de código. Se muestra un trabajo bien organizado en el repositorio.	La aplicación web se encuentra versionada correctamente con GIT en GITHUB, Bitbucket o cualquier otro versionador de código, pero el repositorio puede no estar bien organizado.	La aplicación web NO se encuentra versionada correctamente con GIT en GITHUB, Bitbucket o cualquier otro versionador de código. No se muestra un repositorio.

	Se encuentra un README.md donde se indican claramente los pasos y requerimientos para levantar la aplicación. ***Nota: Si la aplicación no se puede ejecutar siguiendo los requerimientos y pasos establecidos NO se procederá a calificar el taller obteniendo la nota mínima de 1.		
--	---	--	--

Pesos de cada criterio y su evaluación

Componente	Descripción	Peso
Interfaz Web	El sistema web contiene lo requerido, implementando todos los requerimientos indicados en el enunciado.	10%
API	Se construyen todos los endpoints correspondientes para dar funcionalidad a los requerimientos del sistema.	10%
Estructura del Taller	La estructura del proyecto corresponde a proyectos independientes tanto para Backend y para Frontend. Se encuentran en una sola carpeta versionada en GitHub.	5%
Implementación de funcionalidades	Se implementa las funcionalidades solicitadas.	70%
Uso de Git	La aplicación se encuentra versionada correctamente con Git en GitHub o BitBucket. Se evidencia un trabajo organizado en el repositorio. Se encuentra el README.md con las instrucciones para levantar el proyecto.	5%

Good Luck! 😊 🍀