Clientes Web Mobile

Final

Profesor: Santiago Gallino.

Nota:

Esta entrega puede realizarse como parte del trabajo realizado para la tesis, o como un trabajo aparte.

En caso de usar la tesis, no aplican los requerimientos listados a continuación, excepto por la modalidad de entrega.

En caso de realizarse en un trabajo aparte, deberá desarrollarse un sitio de una empresa que provea un servicio a clientes (ej: hosting, e-learning, servicios en la nube como administrador de proyectos o campañas de emailing, etc).

Consigna:

El trabajo debe:

- Ofrecer registro y autenticación de usuarios.
- Tener dos roles diferentes de usuarios (administrador y usuario común).
- Tener además de la página web (el "sitio") tener un panel de administración (el "panel").
- Ser una SPA (Single-Page Application).
- Estar creado como un proyecto de npm.
- Utilizar WebPack para el "bundling" de los archivos.
- Usar Firebase Firestore para el almacenado de información y Firebase Authentication para la autenticación/registro de usuarios.
- Permitir que los usuarios "contraten" servicios (no se requiere de pasarela de pago).

- Implementar correctamente las reglas de seguridad de Firestore.
- Incluir upload de archivos (imágenes).

Importante: Puede usarse, de así preferirlo, Vue o React para el desarrollo del frontend. De usar alguna de esas tecnologías, deberán estar debidamente integradas usando lo que la biblioteca ofrece, y podrán ser sujeto de evaluación. Además, **no** podrán usarse paquetes que faciliten la integración con Firebase o de los elementos requeridos para la evaluación.

El "sitio" debe:

- Tener una "home" que presente el servicio que se ofrece.
- Tener una página de "pricing" ("Precios", por ejemplo) donde se listen las posibles opciones de contratación, obtenidos de Firestore. El usuario debe poder contratar desde acá el servicio.
- Incluir registro/login de usuarios.
- Posibilidad para usuarios autenticados de contactar con el proveedor del servicio vía chat o sistema de mensajes, ambos ofreciendo actualizaciones en tiempo real usando Firestore.
- Tener un "Perfil" para usuarios autenticados, donde puedan ver los mensajes/chats intercambiados con el proveedor, y donde administrar sus datos.
- Desde el “Perfil”, el usuario debe poder cambiar el tipo de suscripción, o cancelarla.

El "panel" debe:

- Solo ser accesible para usuarios con rol "administrador".
- Permitir administrar las opciones de contratación, incluyendo sus detalles.
- Ver los mensajes/chats realizados por clientes, para poder responderlos.
- Ver los usuarios y los servicios que tienen contratados, en caso de tenerlos.

Frontend

El HTML debe ser semántico y estar correctamente estructurado.

Debe incluir estilización en CSS. Puede usarse un framework de CSS como Bootstrap o Tailwind de base. En case de usar Vue o React, puede usarse un paquete de componentes estilizados, pero siendo el alumno/a responsable de asegurarse de que el HTML y su usabilidad, accesibilidad y semántica resultantes sean adecuados.

Backend

No se requiere de un backend. Firebase va a cumplir esas necesidades para esta entrega.

<u>Se evaluará y tendrá impacto en la nota también:</u>

- Complejidad de la tarea realizada.
- **Uso correcto de las etiquetas semánticas de HTML.**
- Aplicación de buenas prácticas acordes a cada tecnología utilizada (principios de programación Orientada a Objetos en php, normalización en SQL, correcta separación de componentes, incluyendo uso apropiado de propiedades, state y eventos en React, etc).
- Coherencia en los nombres de variables, clases, métodos, etc.
- Documentación apropiada usando PHPDoc y JSDoc, según corresponda.
- Estilización del sitio.
- Prolijidad en el código.
- Prolijidad en la organización de la carpeta del proyecto.
- Usabilidad del sitio.

Modalidad de entrega

La entrega se realizará de manera digital, en el DV Panel según sea indicado por la Escuela, en un archivo con los siguientes requerimientos:

- Debe tener el nombre con el formato:
  - **apellido-nombre_final_.[rar|zip]** en caso de ser un solo integrante.
  - **apellido-nombre_apellido-nombre_final_.[rar|zip]** en caso de ser 2 o más integrantes.
  - Ej:
    - **gallino-santiago_final_.rar**
    - **gallino-santiago_noto-federico_final_.rar**
- El archivo de la entrega debe contener:
  - **Todos** los archivos necesarios para correr el proyecto (incluyendo carpetas como **node_modules** o **vendor** para acelerar el proceso de corrección).
  - En caso de usar Firebase, tener las reglas copiadas en un archivo **rules.json**, y un ejemplo de los datos de Firestore en **database.json**.
  - En caso de usar MySQL, tener el **.sql** exportado con el contenido completo de la base, y el DER.
  - Un archivo **datos.txt** con la siguiente información:
    - Carrera, materia, cuatrimestre, año, turno, comisión, apellido y nombre (todos los integrantes), docente, carácter de entrega (final). Además debe incluir las credenciales de acceso al panel de administración.

El incumplimiento de la modalidad de entrega en cualquiera de sus puntos estipulados puede incurrir en una penalización en la nota de *al menos* un punto.

De detectarse un trabajo copiado, tanto copiador como copiado recibirán automáticamente una calificación de 1 (uno).

Quedará a discreción del profesor si es necesario realizar preguntas a los estudiantes, ya sean con respecto a cómo se encaró la entrega, como

teóricas pertinentes a la materia o las tecnologías asociadas a la misma (HTML, CSS, JS, php, MySQL, React, etc).