**Programación con Entornos de Trabajo**

Segundo Parcial

**Profesor: Santiago Gallino.**

**Nota:**

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

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).

**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.

El "panel" debe:

* Solo ser accesible para usuarios con rol "administrador".
* Permitir administrar las opciones de contratación.
* 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.

Se evaluará y tendrá impacto en la nota también:

* 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, dejando una copia del proyecto completo (front y back) en el ftp de entregas en una carpeta con el apellido y nombre de los/as alumnos/as, que contenga:

* un archivo **datos.txt** con todos los siguientes datos:
  + Carrera, materia, cuatrimestre, año, turno, comisión, apellido y nombre, docente, carácter de entrega (2do parcial).
* un archivo **rules.txt** con las reglas de Firestore.

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).