NECESIDAD QUE VAMOS A CUBRIR

Cada día la gente viaja más y planifica sus viajes con más antelación, por lo que, poder guardar ofertas de hoteles y vuelos para poder verlos más adelante o simplemente poder comparar los precios será bastante útil para los viajeros.

IDEA

Crear una aplicación web que me permita filtrar hoteles por precio y guardar en mis favoritos los hoteles y vuelos con la intención de compararlos.

REFERENCIAS

<https://www.tripadvisor.es/>

<https://www.booking.com>

MODELO DE NEGOCIO

Este proyecto lo puedo monetizar mediante publicidad en AdSense o mediante marketing de afiliados (por ejemplo booking)

FUNCIONALIDADES / CARACTERÍSTICAS

* Registro, identificación
* Buscar hoteles por fecha, habitación y número de ususarios
* Filtrar hoteles por precio
* Información de hoteles
* Guardar hoteles en favoritos y que el usuario pueda verlos(solo los suyos) (poder filtrar por precios)
* Búsqueda de vuelos
* Guardar Vuelos en favoritos y que el usuario pueda verlos
* Mostrar noticias en la página principal(scraping)
* Mostrar la programación del tiempo para viajes en los proximos 30 días
* Posibilidad de contactar con nosotros
* Posibilidad de encontrar hoteles en el mapa de google
* Registra tu propio alojamiento
* Multiidioma
* Facilidad de uso (Usabilidad)
* Diseño elegante y limpio (Diseño)

ÁRBOL WEB / CONTENIDOS

* Página principal o de aterrizaje (**pública**)
* Login (**pública**)
  + Registro (**pública**)
  + Solicitar modificar contraseña (**pública**)
  + Modificar contraseña (**pública con token**)
* Zona de usuario (**privada**)
  + Datos principales de la cuenta (**privada**)
* HOTELES (**pública**)
* VUELOS (**pública**)
* MIS FAVORITOS (**privada**)
  + Guardar y filtrado de hoteles por precio
* Aviso legal, política de privacidad, política de cookies… (**pública**)
* Contacto (**pública**)

PERFILES DE USUARIO

* **Usuarios anónimos**: Tendrán acceso a todo el contenido público exclusivamente.
* **Usuarios registrados**: Tendrán acceso a todo el contenido público y además podrán acceder a la página de favoritos, donde tendrán guardados sus hoteles y vuelos.

PROCESOS

Registro

El registro se hará mediante una serie de datos básicos que serán:

* Email
* Contraseña

Una vez que el usuario haya introducido correctamente sus datos, se le enviará un email de bienvenida con un enlace que verificará la creación de la cuenta.

Además, si los datos introducidos son correctos y la cuenta se ha creado, independientemente de que el email se haya verificado o no, el usuario quedará autenticado automáticamente en la aplicación.

Identificación

Se le pedirá al usuario email y contraseña para acceder. Si todo va correctamente se le enviará a la página principal.

Modificación de contraseña

El usuario introducirá la contraseña asociada a su cuenta en nuestra aplicación. Si todo va bien, el usuario recibirá un correo electrónico informando sobre la solicitud de cambio de contraseña con un enlace para comenzar el proceso de modificación de contraseña.

Cuando el usuario pulsa sobre este enlace, accederá a una pantalla pública en la que deberá introducir la nueva contraseña (siguiendo las buenas prácticas en materia de seguridad)  y deberá pulsar en el botón modificar.

Si la contraseña se modifica correctamente, se le enviará otro email informando sobre el cambio de contraseña.

Confección de búsqueda de hoteles

Tendré una caja de búsqueda en la que se recogerán los siguientes datos:

* Ciudad
* Fecha de checkin
* Noches
* Nº Adultos
* Nº de habitaciónes

Mediante peticiones a una api se pintarán los datos de los hoteles en una caja justo debajo de la caja de búsqueda, estos datos son:

* título del hotel
* foto
* categoría
* precio
* localización
* votos
* numero de comentarios

--------------------------------------------------------------

ARQUITECTURA SOFTWARE

**API REST**: Se encargará de servir y recibir las peticiones de información de los hoteles.

TECNOLOGÍAS

API REST

**Tecnologías**

* **NODE** - Entorno de ejecución de JavaScript (Base)
* **MONGO** - Persistir datos en el disco

**Dependencias producción**

* **EXPRESS** - Framework de desarrollo
* **BEARER TOKEN** - Permite extraer de la cabecera Authorization el token bearer que se le pase (si no se le pasa ninguno, en la petición tendrá valor nulo). El token estará almacenado en req.token (dentro del middleware)
* **BODY PARSER** - Permite convertir el cuerpo de una petición a formato JSON automáticamente
* **JWT** - Permite crear y validar un token JWT.
* **CORS** - Permite que se pueda llamar a mi API desde cualquier sitio (pudiendo si quiero restringir desde donde se llama)
* **MONGOOSE** - Crear esquemas y modelos para definir los modelos de datos en Mongo

**Dependencias desarrollo**

* **NODEMON** - Dependencia de desarrollo para poder utilizar la funcionalidad hot reloading.
* **DOTENV** - Para precargar variables de entorno desde fichero .env

**Sistemas**

* Sistema Linux (distro Ubuntu) 20 x64
* Servidor Node 10+ LTS
* Servidor Mongo 4+ LTS
* SSL Let's Encrypt gratuito

CLIENTE CV ONLINE

**Tecnologías**

* **NUXT** - Framework de desarrollo en el lado del cliente con SSR

**Dependencias producción**

* **AXIOS** - Librería para realizar llamadas HTTP
* **BOOTSTRAP VUE** - Framework maquetación para montar interfaces responsive basadas en Bootstrap
* **PWA** - Habilitar funcionalidades de Progressive Web Application
* **JWT-DECODE** - Decodificación del token JWT
* **VUE-i18N** - Librería para trabajar con internacionalización

**Dependencias desarrollo**

* **SASS** - Precompilador de CSS (mejoras futuras)
* **DONTENV** - Carga de variables de entorno en Nuxt

PLANIFICACIÓN

**Tiempo de entrega**: 3 semanas (**completas**)

Bloques a desarrollar:

1. Frontend (Cliente CV Online)
   1. Páginas públicas (página principal, contacto, preguntas frecuente, plantillas y guía confección cv…)
   2. Registro, identificación, solicitar modificar contraseña y modificar contraseña
   3. Proceso de búsquedas de hoteles
2. Backend (API REST)
   1. Endpoints asociados a páginas públicas (información página principal, contacto, plantillas y búsquedas de hoteles…)
   2. Endpoints registro, login, logout, solicitar modificar contraseña, modificación contraseña
   3. Endpoints para guardar y eliminar hoteles.
   4. Endpoints para los filtros

ORGANIZACIÓN DEL ENTORNO DE TRABAJO

Cómo me voy a organizar:

* Editor: **Visual Studio** con las siguientes extensiones:
  + Git
  + DotEnv
  + EditorConfig
  + Rainbow Brackets
  + Visual Studio IntelliCode

Otras consideraciones por repositorio

* Fichero **.editorconfig** en el raiz de cada proyecto
* Ficheto **.gitignore** en el raiz de cada proyecto (ignorado cualquier dato sensible, dependencia, etc.)
* **README.me** con instrucciones de como instalar por primera vez el proyecto
* **Rama master** para versiones en producción (estables) (tageadas con su versión correspondiente)
* **Rama develop** para versiones en desarrollo (inestables)