

Desarrollo de Aplicaciones Web



Florida

Universitària

Proyecto Final de Ciclo de Desarrollo de Aplicaciones Web

Título: GUEXT1080

Autor: Alberto Villegas Chaparro

**Curso Académico: 2º Ciclo Superior de Desarrollo de
Aplicaciones Web**

ÍNDICE

APARTADOS PROPUESTOS.....	1
1. RESUMEN DEL PROYECTO.....	1
2. JUSTIFICACIÓN Y OBJETIVOS DEL PROYECTO.....	3
3. DESARROLLO DEL PROYECTO.....	3
a. Análisis del mercado y posible modelo de negocio.....	3
b. Metodologías utilizadas	5
c. Descripción de los componentes de la aplicación.....	7
d. Problemas/dificultades encontradas	13
e. Resultados obtenidos.....	16
4. CONCLUSIONES	21
5. LÍNEAS FUTURAS DE TRABAJO	22
6. WEBGRAFÍA.....	22
7. ANEXOS	26
1. ANEXO 1. GUÍA DE ESTILOS	27

APARTADOS PROPUESTOS

1. RESUMEN DEL PROYECTO

La aplicación trata sobre una integración de un editor HTML de tipo WYSIWYG llamado Summernote en GUEXT1080, una aplicación proporcionada por la empresa en la que se incluirá nuevas funcionalidades con el fin de crear y gestionar plantillas para el envío de correos electrónicos. También, les permitirá editar o visualizar el contenido sin necesidad de escribir código manualmente, por lo que será una manera de trabajar muy cómoda y rápida.

Otras tareas que se llevarán a cabo son que existirán varias versiones de la misma plantilla en distintos idiomas permitiendo cambiarlas fácilmente cuando sea necesario. Asimismo, estas plantillas se podrán personalizar añadiendo los estilos deseados e insertar imágenes o varios placeholders de variables según el contexto que se haya seleccionado sin problemas. Y después, los usuarios podrán previsualizar estas plantillas con las variables que estarán sustituidas por los valores correspondientes de la base de datos.

Adicionalmente, habrá un apartado de filtrado para mostrar únicamente los registros específicos de las plantillas, lo que dejará a los usuarios encontrar con rapidez la plantilla que están buscando.

Con respecto a las tecnologías, en el proyecto se usarán para la parte del frontend, el framework React con el lenguaje de programación JavaScript y aparte de esto, CSS y HTML. En cuanto al backend, se empleará Symfony con PHP creando una API REST a través de API Platform y para gestionar los datos, se integrará Doctrine ORM y la base de datos se implementará en MySQL.

Abstract:

The application is about an integration of a WYSIWYG type HTML editor called Summernote in GUEXT1080, an application provided by the company that will include new functionalities to create and manage templates for sending emails. Also, it will allow you to edit or view the content without having to write code manually, so it will be a very comfortable and fast way of working.

Other tasks that will be carried out are that there will be several versions of the same template in different languages allowing them to be easily changed when necessary. These templates can also be customized by adding the desired styles and inserting images or several variable placeholders according to the context that has been selected without problems. And then, users will be able to preview these templates with the variables that will be replaced by the corresponding values of the database.

Additionally, there will be a filtering section to show only the specific records of the templates, which will let users quickly find the template they are looking for.

Regarding technologies, the project will use the frontend part, the React framework with JavaScript programming language and apart from this, CSS and HTML. As for the backend, Symfony will be used with PHP by creating a REST API through API Platform and to manage the data, Doctrine ORM will be integrated, and the database will be implemented in MySQL.

2. JUSTIFICACIÓN Y OBJETIVOS DEL PROYECTO

El proyecto está dirigido a los usuarios de una plataforma de gestión hotelera bastante compleja de la empresa para crear y administrar plantillas de correos electrónicos. Se propone ya que actualmente, este proceso se realiza manualmente mediante un editor de código, lo que lo convierte en una tarea lenta, compleja y propensa a errores humanos, además de requerir conocimientos técnicos.

A través de estas funcionalidades incorporadas, se pretende dar una herramienta visual e intuitiva que permita realizar las siguientes tareas:

- Facilitar la creación y gestión de plantillas sin conocimientos técnicos.
- Permitir la gestión multilingüe de plantillas.
- Mejorar la eficiencia y reducir errores en el proceso de envío de emails.

3. DESARROLLO DEL PROYECTO

- a. Análisis del mercado y posible modelo de negocio: qué propuestas similares hay, cuál es el valor añadido de esta propuesta, cómo se podría explotar, etc.

Actualmente, encontramos distintas aplicaciones en el mercado que se encargan de la gestión de housekeeping y mantenimiento para hoteles. Estas soluciones sirven para agilizar los procesos internos, facilitar la organización del equipo de trabajo y asegurar una experiencia adecuada tanto para el personal como para los huéspedes.

A continuación, se presentan algunas de las más destacadas que son:

- **HKeeper:** Es una plataforma completa para la gestión de tareas y personal en hoteles. Proporciona funcionalidades como conserje virtual, mantenimiento, listas de verificación y comunicación interna. Además, reduce los gastos de forma considerable y centraliza todas las operaciones en un único sistema.
- **SabeeApp:** Es una solución alojada en la nube que da herramientas para gestionar housekeeping, mantenimiento, reservas y facturación. También, posibilita la organización de las tareas eficazmente, favorece una comunicación fluida y permite la elaboración de informes y la accesibilidad para visualizar la información desde cualquier dispositivo.
- **Hotelkit:** Es una plataforma en la nube que permite la gestión de housekeeping, mantenimiento y recepción en hoteles. También, ayuda a la asignación de tareas de forma instantánea, la comunicación rápida y sencilla entre departamentos y el uso de listas de verificación digitales.

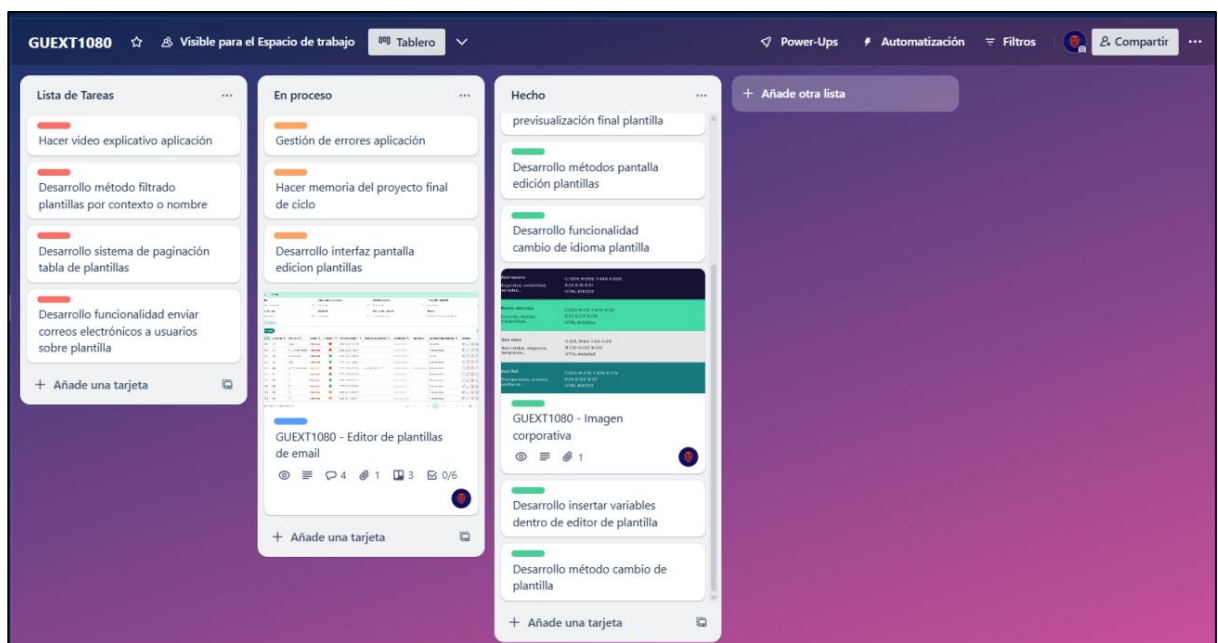
Asimismo, deja actualizar el estado de las habitaciones en tiempo real, documentar procesos y optimizar la eficiencia operativa con su integración con PMS.

El valor añadido de la propuesta se resume en su integración directa con los sistemas internos del hotel, lo que simplifica la optimización de procesos y da una personalización avanzada en las preferencias del cliente. Mediante el conocimiento de sus preferencias, se presentan soluciones adaptadas y específicas que aumenta la experiencia del usuario de forma importante.

b. Metodologías utilizadas

Las metodologías que utilizaré son la metodología ágil SCRUM para la gestión del proyecto, un marco de trabajo que ayuda a los equipos a organizarse de manera autónoma y trabajar en ciclos de desarrollo cortos nombrados sprints con el fin de entregar las tareas propuestas.

Para ello se usará Trello, una herramienta para hacer un seguimiento del progreso de las tareas de manera eficaz. A continuación, se presenta una imagen sobre cómo se ha realizado el proceso.



Además de esto, se irá detallando que es lo que se ha ido haciendo a lo largo de las reuniones de Sprint mediante videollamada:

25/03/2025 – Primera reunión del equipo de desarrollo de proyecto final de ciclo

Hemos establecido la primera entrega de los siguientes elementos que son el wireframe, la estructura de la base de datos de la aplicación y también, parte de la guía de estilos realizada.

17/04/2025 – Segunda reunión del equipo de desarrollo de proyecto final

Hemos corregido las tareas que habíamos entregado paso a paso para tener una idea de cómo realizar los documentos y la memoria adecuadamente y así, progresar con nuestro proyecto final de ciclo.

06/05/2025 - Tercera reunión del equipo de desarrollo de proyecto final

En esta reunión, hemos estado resolviendo dudas sobre la aplicación para tenerlo todo claro y saber en qué situación nos encontramos en estos momentos. Además, nos ha detallado algunas cosas que hay que tener en cuenta de cara a la documentación y al proyecto aplicativo.

15/05/2025 - Cuarta reunión del equipo de desarrollo de proyecto final

En este último sprint, hemos aclarado algunas dudas con respecto a la memoria del proyecto. También, nos hemos dedicado la mayoría de tiempo a preparar la presentación y la defensa del PFC comentando lo que hay que hacer y lo que hay que evitar, además de algunas recomendaciones que hay que tener en cuenta.

- c. Descripción de los componentes de la aplicación: mockups, arquitectura, *backend*, tecnologías utilizadas, etc.

1) ANÁLISIS DE REQUISITOS

1.1 Aspectos del Sistema

Las funcionalidades que abarca esta aplicación son:

- Edición visual de plantillas HTML.
- Gestión de plantillas ya sea para crear, editar o eliminar.
- Visualización del contenido generado estructurado en formato tabla.
- Filtrado de información de plantillas en tiempo real.
- Exportación del contenido en código HTML de manera organizada.

1.2 Requisitos Funcionales

➤ Visualización contenido plantillas

Se listará la información de las plantillas en formato tabla para que los usuarios puedan leerlo todo de una manera estructurada y flexible.

➤ Creación nueva plantilla

Creación de nuevas plantillas a través de un editor de plantillas con elementos predeterminados y opciones disponibles como menús desplegables, texto, imágenes, etc.

➤ **Edición plantilla existente**

Los usuarios tendrán la opción de seleccionar una plantilla existente dependiendo del contexto que sea y poder editarla como ellos deseen.

➤ **Filtrado dinámico de plantillas**

Filtración de información de plantillas por contexto o por su nombre mostrando esos registros específicos automáticamente sin necesidad de pulsar un botón.

➤ **Eliminación plantilla**

Los usuarios tendrán la posibilidad de eliminar alguna plantilla desde la sección de listado de plantillas.

➤ **Previsualización final de plantilla**

Habrà una opción para ver una vista previa de la plantilla antes de enviarla y así, comprobar su resultado final con su formato correspondiente, estilos, etc.

1.3 Requisitos No Funcionales

- **Usabilidad**

- La interfaz debe ser intuitiva y adaptada a todo tipo de usuarios, sobre todo para los no técnicos.
- Gestión de errores, que tendrán que estar manejados de manera apropiada a través de mensajes de cara a los usuarios. También, se usarán iconos a modo de ayuda para dar una mayor claridad.

- Diseño responsive ajustado para que se pueda visualizar en diferentes dispositivos como móviles, tablets y ordenadores.

- **Rendimiento**

- La aplicación debe responder de manera rápida en cada uno de los dispositivos. Además, deberá de cargar varias llamadas a la API sin cometer errores.

- **Escalabilidad**

- El sistema estará preparado para añadir nuevas funcionalidades sin deteriorar el rendimiento actual del sistema.

- **Accesibilidad**

- Se tendrá en cuenta unos principios básicos de accesibilidad para los usuarios que tengan distintos tipos de discapacidad como las visuales y auditivas.

1.4 Despliegue

- Servidores virtuales de Linux en la nube para garantizar escalabilidad, disponibilidad y flexibilidad.
- Servidor web llamado Apache2 que se encargará de gestionar el backend del proyecto realizado en PHP. Es el servidor más apropiado porque sirve para aplicaciones PHP o estáticas.

1.5 API con endpoints

- En este proyecto no se ha implementado la autenticación de usuarios, ya que esta parte pertenece al entorno de producción de la empresa. Allí, se emplean tokens JWT con fecha de caducidad, que permiten a los usuarios autenticarse.

Es un sistema basado en roles y permisos que establece qué funciones puede realizar cada usuario autenticado para llamar a los diferentes endpoints de la API.

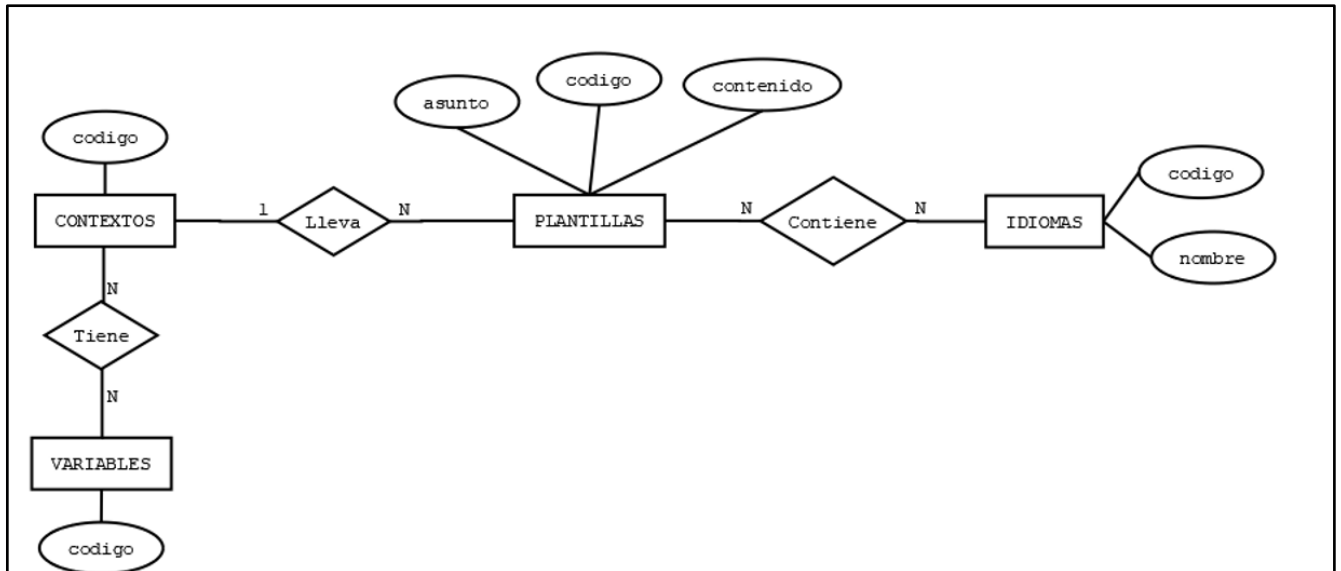
2) ARQUITECTURA

En cuanto a mi proyecto propuesto, está implantado mediante una arquitectura cliente-servidor en el que el cliente que en mi caso será la parte del **frontend** enviará peticiones al servidor para obtener su información o hacer otras operaciones.

Por otro lado, el **backend** que se ocupará de procesar las solicitudes y devolver la respuesta necesaria de cada petición, gestionando la lógica de negocio y acceso a los datos. Gracias a esto, nos ayuda a conseguir una estructura organizada y separada por tareas, lo que facilita la escalabilidad, el mantenimiento y el correcto funcionamiento de la aplicación.

3) DIAGRAMA MODELO ENTIDAD-RELACIÓN DE BBDD

Ahora, se mostrará el modelo de la estructura de la base de datos de la aplicación.



Respecto al diagrama, se comprueba que hay diversas entidades. La primera es **contextos** que representa el tipo de plantilla y está asociada con muchas plantillas y variables. Mediante esto, las plantillas estarán formadas por un único contexto y las variables pueden tener muchos contextos.

La siguiente es **plantillas** que serían los mensajes que se mandan con su asunto y contenido. Su relación pertenece a un contexto únicamente y puede contener muchos idiomas. Por otro lado, tenemos las **variables** que son textos que se sustituirán una vez se haya enviado y están asociadas a muchos contextos, ya que puede haber varias variables en otros contextos.

Por último, la entidad **idiomas** que consiste en traducir la plantilla al lenguaje correspondiente según el usuario desee y se relaciona con muchas plantillas.

4) TECNOLOGÍAS UTILIZADAS

- Backend:

Respecto a la parte del backend, como tecnología principal se ha usado Symfony de PHP desarrollado API REST de simulación de forma manual en el que me ha dejado probar las funciones durante el proceso de implementación. De cara al entorno de preproducción de la empresa, se utilizará Java junto con Spring Boot.

Para la gestión de datos del proyecto, se integrará Doctrine ORM con una base de datos MySQL, donde se almacenarán la información necesaria ya sea las variables, plantillas, etc. Asimismo, se implementarán operaciones CRUD para gestionar dichos datos.

Una vez completado esto, se pasará todo a preproducción.

- Frontend:

▪ JavaScript y React

Para el frontend, se aplicará el lenguaje de programación JavaScript para desarrollar el código de la aplicación.

También, se empleará el framework React, que nos permitirá crear una interfaz de usuario interactiva y dinámica, aumentando la experiencia del usuario con componentes reutilizables y actualizaciones rápidas.

- **HTML y CSS**

Adicionalmente, se usará lenguajes de marcado que nos ayudará a mejorar en la parte visual del proyecto. En primer lugar, HTML que nos dejará organizar y estructurar el contenido de forma lógica y responsiva.

Después, CSS que servirá para dar estilos a los elementos del proyecto y de esta manera, llamar la atención a los usuarios proporcionando un diseño atractivo y coherente.

- **Bootstrap**

Para la aplicación, se empleará Bootstrap, un framework de CSS que posibilita el desarrollo de páginas web responsivas para realizar un diseño que sea adaptado a diferentes dispositivos como móviles, ordenadores, etc.

d. Problemas/dificultades encontradas en el desarrollo del PFC, y que soluciones se han buscado para su solución

A lo largo de estos últimos meses de desarrollo del PFC me he encontrado con diversos problemas o errores que me han resultado ser complicados a la hora de resolverlos, pero con el paso de días se ha ido resolviendo las cosas y obteniendo resultados óptimos para la aplicación.

Los problemas o dificultades que he encontrado en el desarrollo del PFC han sido cuando tenía que manejar distintas versiones de plantillas en varios idiomas, ya que había que interpretar el contenido de forma diferente según el idioma, por lo que era más complicado.

Para solucionarlo, guardé los datos en formato JSON como tipo texto en la base de datos y así, leería una información u otra dependiendo del código del idioma de manera más eficiente. Además, al crear las plantillas, debía tener en cuenta que, si en el futuro se añadían nuevos idiomas, debía de estar reflejado. Por ello, una vez que iba añadiendo un idioma con sus atributos correspondientes, se iba incorporando progresivamente en las plantillas.

Aquí observamos un ejemplo de código de como se ha gestionado los idiomas cuando cambio de plantilla.

```
const DropdownTemplate = ({
  const handleLanguageChange = async (langDropdown) => {

    if (selectedTemplate) {
      setLoadingEditor(true);
      const response = await listTemplateById(selectedTemplate.id, setAlert, setVisibleAlert);

      if (isTemplateModified()) {
        setWarningMessage("Estás seguro de que deseas cambiar de idioma? Se perderán los cambios.");
        setTextButton("Cambiar idioma");
        setVisibleModalWarning(true);

        setConfirmAction(() => () => {
          if (response) {
            setSelectedTemplateContent(response.data[selectedLanguage.code].content);
            setNameTemplate(response.code);
            setSubjectTemplate(response.data[selectedLanguage.code].subject);
            setOriginalSubjectTemplate(response.data[selectedLanguage.code].subject);
            setSelectedLanguageDropdown(selectedLanguage.value);
            setCodeLanguage(selectedLanguage.code);
          } else {
            resetData();
            setSelectedLanguageDropdown(selectedLanguage.value);
            setCodeLanguage(selectedLanguage.code);
          }
        });
      } else {
        setSelectedTemplateContent(response.data[selectedLanguage.code].content);
        setSubjectTemplate(response.data[selectedLanguage.code].subject);
        setOriginalSubjectTemplate(response.data[selectedLanguage.code].subject);
        setSelectedLanguageDropdown(selectedLanguage.value);
        setCodeLanguage(selectedLanguage.code);
      }
    } else {
      setSelectedLanguageDropdown(selectedLanguage.value);
      setCodeLanguage(selectedLanguage.code);
    }
  }
});
```

Otros problemas que han aparecido han sido cuando estaba desarrollando la función para sustituir los valores de placeholders de las variables por datos reales de la base de datos, ya que las variables dependían del tipo de contexto. Al principio, no entendía correctamente como era la lógica para realizar esta parte, pues estuve durante varias horas pensando cómo hacerlo de manera adecuada y con sentido.

Para su solución, con la ayuda de la empresa pude lograr el objetivo y lo primero que había que pensar es que, en este proyecto la idea de usar contextos fue para agrupar y obtener un conjunto de datos distintos según cada caso y así, hacer el reemplazo en las variables de las plantillas adecuadamente.

Por lo tanto, desde el frontend en el cuerpo de la petición se introdujo identificadores de datos como usuarios, huéspedes, hoteles a modo de simulación y después, en el backend se hizo una llamada para que según el contexto que hayamos seleccionado, obtenemos los datos de la base de datos dependiendo de los identificadores escogidos.

De esta manera, he podido implementar varios ejemplos de plantillas de distintos tipos de contexto, en los que se puede comprobar que los valores se sustituyen por datos que hay en las tablas compuestas de la base de datos.

A continuación, se muestra una imagen como ejemplo de cómo se ha realizado el proceso de reemplazo de placeholders por valores reales.

```
final class PlantillasController extends AbstractController
public function renderTemplate(Request $request): JsonResponse

    $variables = $context->getVariables();
    $contextCode = $context->getCode();

    $replacementValues = $this->showDataPlaceholders(contextCode: $contextCode, data: $data);

    if (isset($replacementValues['error'])) {
        return new JsonResponse(data: ['error' => $replacementValues['error']], status: 400);
    }

    foreach ($variables as $variable) {
        $code = $variable->getCode();
        $placeholder = '{{' . $code . '}}';
        if (strpos(haystack: $content, needle: $placeholder) !== false || strpos(haystack: $subject, needle: $placeholder) !== false) {
            $value = $replacementValues[$code];
            $content = str_replace(search: $placeholder, replace: $value, subject: $content);
            $subject = str_replace(search: $placeholder, replace: $value, subject: $subject);
        }
    }

    return new JsonResponse(data: [
        'rendered' => $content,
        'subject' => $subject
    ]);
}
```

Por último, en la parte de edición de plantillas tenía un inconveniente que, si el usuario había hecho cambios en la plantilla seleccionada, había que controlar esos cambios para advertirle y sino, que se pudiera cambiar sin ningún tipo de problema.

Como solución, lo que propuse fue que tenía que guardar una variable con el contenido que me había guardado en la base de datos y otra cogiendo lo que estaba introduciendo recientemente. Además, como el editor Summernote guardaba la información en código HTML era más difícil, porque había que verificar si tenía espacios y textos iguales. Finalmente, realicé una función que permitía colocar el código en formato texto limpio y así, comprobar de manera clara si los dos contenidos eran semejantes.

e. Resultados obtenidos

Una vez implementada la aplicación con varias funcionalidades completas y eficaces, los resultados han sido óptimos, ya que se han cumplido todas las expectativas y objetivos que me había propuesto. Una de las partes más importantes del proyecto ha sido la edición de plantillas, donde el usuario puede gestionarlas como desee según el idioma, añadiendo los estilos que considere y las variables necesarias para que la plantilla esté lo más correcta posible. De esta forma, será de mucha ayuda de cara al usuario y podrá editar cualquiera con mucha rapidez.

En este proyecto no se ha realizado y gestionado un sistema de autenticación de usuarios, ya que es un entorno de pruebas y una incorporación a la aplicación de la empresa.

Generalmente, he logrado resultados que no me esperaba, ya que pensaba que no llegaría a realizar tantas cosas como tenía previsto el primer día debido a la falta de tiempo.

A continuación, explicaremos como navegar por las diferentes pantallas de la aplicación web. Una vez entre el usuario, observará una cabecera con el logo, un nombre ficticio de la empresa y un menú de opciones. Más abajo, tenemos la sección de filtrado en el que aparecerán las plantillas según el contexto o nombre.

Después, está la tabla de listado de plantillas en el que podrán visualizarlos todos los usuarios o solo uno cada una de ellas y también, eliminarlas o acceder a su contenido para editarlas o previsualizarlas cuando sea necesario.

LISTADO DE PLANTILLAS

Filtros

Contextos

Seleccionar...

Buscar

Nombre

Limpiar

+ Crear

Id ↑↓	Contexto ↑↓	Nombre ↑↓	Asunto	Contenido	Acciones
1	BOOKING_MANAGEMENT	BOOKING_CONFIRMATION	dede	DEDEdedede	<div><div></div><div></div><div></div></div>
				Estimado {{USER_NAME}} {{USER_SURNAME}};¡Bienvenido a {{HOTEL_NAME}}! Nos complace que nos haya elegido para su estancia. Nuestro equipo se dedica a garantizarle una experiencia cómoda y placentera. Si necesita algo, nuestra recepción está aquí	<div><div></div><div></div><div></div></div>

+ Crear

Id ↑↓	Contexto ↑↓	Nombre ↑↓	Asunto	Contenido	Acciones
1	BOOKING_MANAGEMENT	BOOKING_CONFIRMATION	dede	DEDEdedede	<div><div></div><div></div><div></div></div>
2	USER_MANAGEMENT	WELCOME_EMAIL	¡Bienvenido a {{HOTEL_NAME}}!	Estimado {{USER_NAME}} {{USER_SURNAME}}, ¡Bienvenido a {{HOTEL_NAME}}! Nos complace que nos haya elegido para su estancia. Nuestro equipo se dedica a garantizarle una experiencia cómoda y placentera. Si necesita algo, nuestra recepción está aquí para ayudarle. Aproveche al máximo nuestras comodidades y esperamos que su estancia con nosotros sea inolvidable. Gracias por elegir {{HOTEL_NAME}} como su hogar lejos del hogar. Atentamente,{{HOTEL_NAME}}	<div><div></div><div></div><div></div></div>

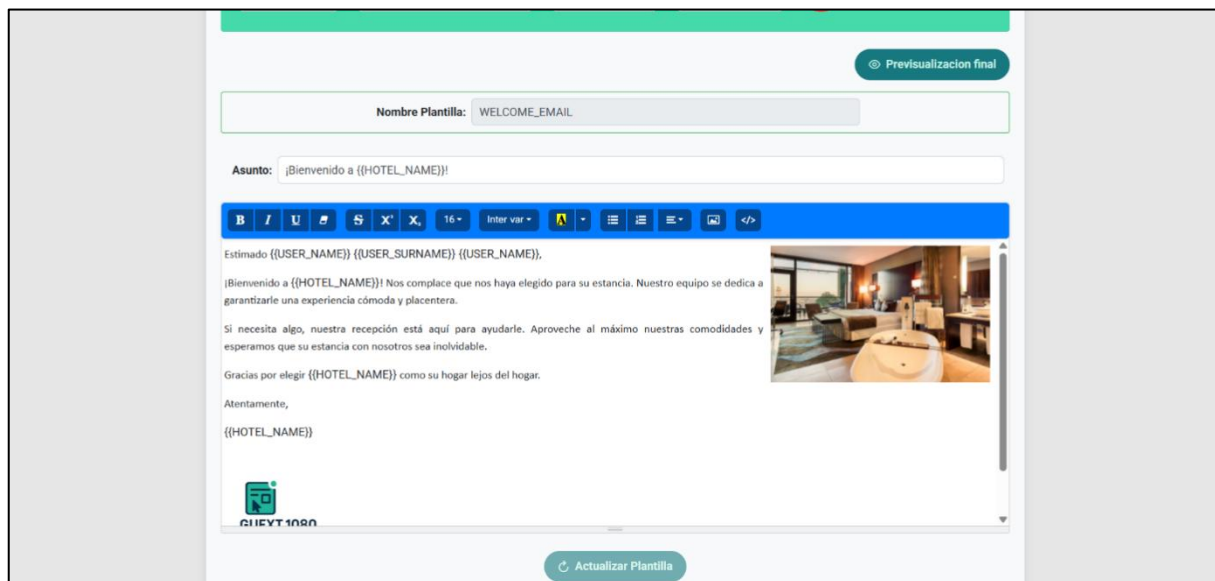
Mostrando 2 plantillas en total

<< < 1 > >>

5 ▾

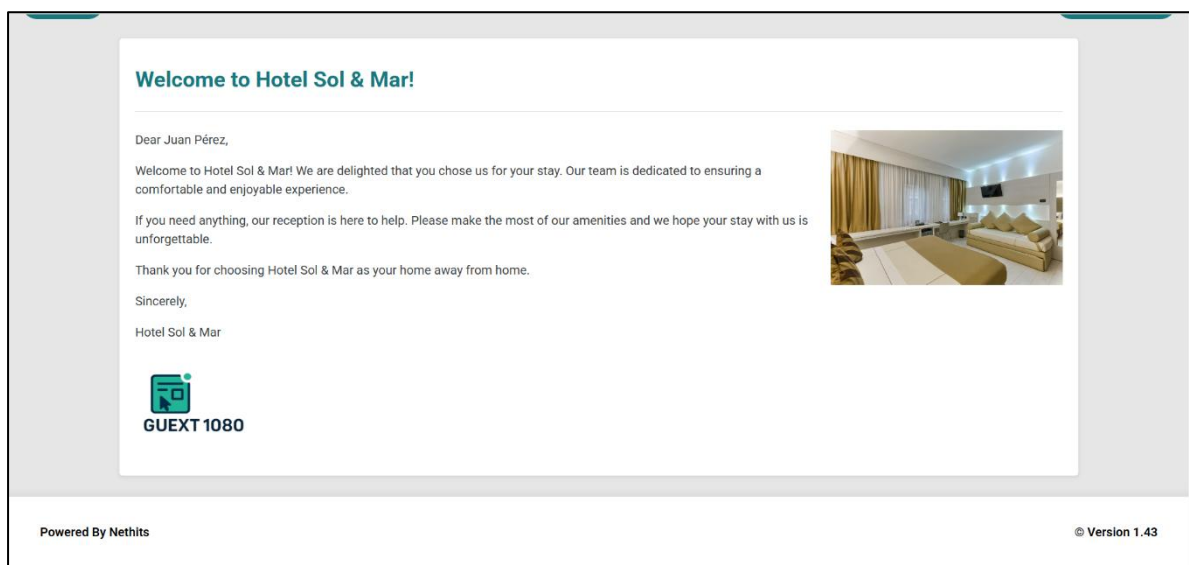
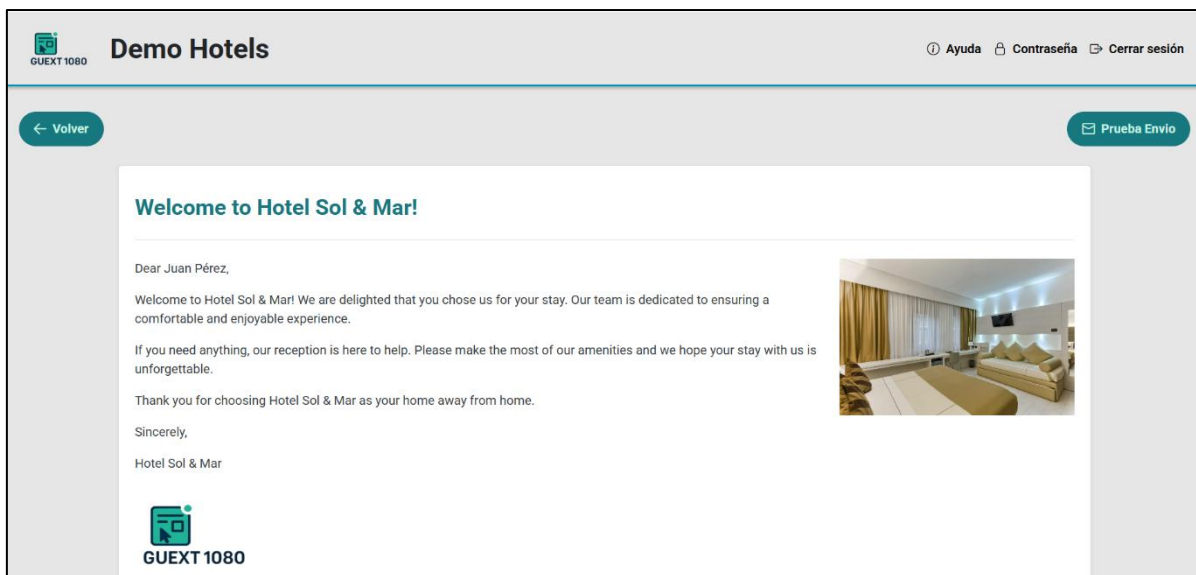
Además, tenemos el pie de página en el que vemos que hay dos textos, uno en el lado izquierdo y otro en el lado derecho, con la versión del sistema y el logo de copyright para los derechos de autor.

Una vez vayamos a editar la plantilla, accederemos a la pantalla que vemos en la imagen en el que observaremos que hay diferentes menús desplegables que se usarán a modo de ayuda para intercambiar de idioma, contextos, plantillas y variables. Más abajo, tendremos la plantilla que hemos pulsado con su asunto y contenido y también, nos permite previsualizarla para que comprobemos como ha quedado.



En esta parte, tenemos la vista previa de la plantilla con los valores sustituidos de los placeholders de variables por los de la base de datos. Como podemos visualizar, está formado por asunto y contenido que ha editado el usuario con una imagen en caso de que lo haya insertado.

Además, disponemos de una opción para volver a la plantilla para editarla de nuevo y otra para enviarla a un correo a modo de prueba al destinatario que escojamos.



4. CONCLUSIONES

Conclusiones: conclusiones obtenidas tras la realización del PFC, tanto a nivel profesional como personal

A mi parecer, gracias a la realización de este proyecto he adquirido muchas cosas que me han ayudado a realizar un trabajo grandioso y original. Una de las cosas ha sido tener autosuficiencia en mí mismo a la hora de solucionar los problemas que he tenido a lo largo de estos meses, con ayuda de documentación y páginas web. Adicionalmente, al trabajar por mi propia cuenta, he conseguido muchos conocimientos que antes no sabía con el framework y el lenguaje de programación que estaba manejando.

Durante los primeros días no sabía si esta idea era adecuada para el proyecto, ya que era una aplicación muy costosa y no sabía si iba a llegar a tiempo para entregarlo. El proceso de adaptación ha sido complicado, pero con empeño y actitud se han ido sacando y cumpliendo las tareas como se requerían.

Asimismo, con la colaboración de la empresa y las recomendaciones que me proporcionaban, he alcanzado conocimientos a nivel profesional que no sabía con anterioridad a través de enlaces interactivos de documentación y archivos.

En conclusión, este proyecto me ha permitido cumplir con las expectativas que me había propuesto al inicio y me ha aportado una ayuda tanto personal como profesional que me ha servido para implementar una aplicación adaptada a todos los usuarios.

5. LÍNEAS FUTURAS DE TRABAJO

En este apartado, se comentarán las líneas futuras de trabajo del proyecto y las posibles implementaciones que se pueden llegar a desarrollar. A continuación, se detallan con profundidad:

- Almacenar las imágenes desde el servidor backend.
- Tener permisos en los usuarios para las funciones de crear, editar o eliminar plantillas.
- Configurar la tipografía dependiendo de lo que desee el cliente.

6. WEBGRAFÍA

- Verónica Ruiz. (s.f.). Cómo plantear la estructura de navegación de un website. Recuperado 14 mayo 2025, desde <https://www.veronicaruiz.es/como-plantear-la-estructura-de-navegacion-de-un-website/>
- Bootstrap. (2021). Introduction – Bootstrap 4.6. Recuperado 14 mayo 2025, desde <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
- Bootstrap. (2020). Borders – Bootstrap 4.4. Recuperado 14 mayo 2025, desde <https://getbootstrap.com/docs/4.4/utilities/borders/>
- PrimeReact. (s.f.). DataTable – Advanced Filter. Recuperado 14 mayo 2025, desde https://primereact.org/datatable/#advanced_filter
- FreshCart. (s.f.). Homepage 5 (Next.js Demo). Recuperado 14 mayo 2025, desde <https://freshcart-next-js.vercel.app/homepage-5>

- PrimeReact. (s.f.). Icons – PrimeReact. Recuperado 14 mayo 2025, desde <https://primereact.org/icons/>
- Summernote. (s.f.). Summernote en GitHub. Recuperado 14 mayo 2025, desde <https://github.com/summernote/summernote>
- Summernote. (s.f.). Editor Summernote. Recuperado 14 mayo 2025, desde <https://summernote.org/>
- Summernote. (s.f.). Summernote Deep Dive. Recuperado 14 mayo 2025, desde <https://summernote.org/deep-dive/>
- Google Fonts. (s.f.). Roboto – Google Fonts. Recuperado 14 mayo 2025, desde <https://fonts.google.com/specimen/Roboto>
- SabeeApp. (s.f.). Aplicación de limpieza para hoteles. Recuperado 14 mayo 2025, desde <https://www.sabeeapp.com/housekeeping-app>
- Hnext. (s.f.). Hnext AI. Recuperado 14 mayo 2025, desde <https://hnext.ai/>
- Hotelkit. (s.f.). Soluciones para Housekeeping. Recuperado 14 mayo 2025, desde <https://hotelkit.net/solutions/housekeeping/>
- Hotelkit. (s.f.). Productos – Housekeeping. Recuperado 14 mayo 2025, desde <https://hotelkit.net/products/housekeeping/>
- Atlassian. (s.f.). Agile vs Scrum – Atlassian. Recuperado 14 mayo 2025, desde <https://www.atlassian.com/agile/scrum/agile-vs-scrum>
- Doctrine Project. (2023). Query Builder – Doctrine ORM 3.3. Recuperado 14 mayo 2025, desde <https://www.doctrine-project.org/projects/doctrine-orm/en/3.3/reference/query-builder.html>
- PrimeReact. (s.f.). DataTable – PrimeReact. Recuperado 14 mayo 2025, desde <https://primereact.org/datatable/>

- Mozilla. (s.f.). String.prototype.trim() – JavaScript | MDN. Recuperado 14 mayo 2025, desde https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/String/trim
- React. (s.f.). useRef – React Docs. Recuperado 14 mayo 2025, desde <https://react.dev/reference/react/useRef>
- Mozilla. (s.f.). Using Media Queries – CSS | MDN. Recuperado 14 mayo 2025, desde https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries
- Mhnpd. (s.f.). React Loader Spinner – Documentación. Recuperado 14 mayo 2025, desde <https://mhnpd.github.io/react-loader-spinner/docs/intro>
- Hackernoon. (s.f.). Guía paso a paso para pantallas de carga en React. Recuperado 14 mayo 2025, desde <https://hackernoon.com/lang/es/guia-paso-a-paso-para-crear-3-diferentes-tipos-de-pantallas-de-carga-en-react-lu2633nd>
- YouTube. (s.f.). Video sobre pantallas de carga en React. Recuperado 14 mayo 2025, desde <https://www.youtube.com/watch?v=DO-pSysGltQ>
- Canary Technologies. (s.f.). 14 Plantillas de correos para huéspedes. Recuperado 14 mayo 2025, desde <https://www.canarytechnologies.com/es/post/14-hotel-guest-email-templates>
- Holidu. (s.f.). Correos electrónicos para huéspedes. Recuperado 14 mayo 2025, desde <https://www.holidu.es/magazine/correos-electronicos-para-huespedes>
- One.com. (s.f.). Ejemplos de correos electrónicos profesionales. Recuperado 14 mayo 2025, desde <https://www.one.com/es/correoelectronico/ejemplos-de-correos-electronicos-profesionales>

- Sirvoy. (s.f.). Textos modelo para correos electrónicos a huéspedes. Recuperado 14 mayo 2025, desde <https://sirvoy.es/blog/topic/maneje-sus-reservas/plantillas-de-mensajes-y-mensajeria-automatizada/textos-modelo-para-correos-electronicos-enviados-a-los-huespedes/>
- Lodgify. (s.f.). Plantillas de correo electrónico para anfitriones. Recuperado 14 mayo 2025, desde <https://www.lodgify.com/blog/es/plantillas-correo-electronico/>
- Freepik. (s.f.). Editor de imágenes Pikaso – Confirmación de reserva. Recuperado 14 mayo 2025, desde <https://www.freepik.es/pikaso/image-editor?prompt=confirmacion+de+reserva&submit=1&style=noStyle&sign-up=google&creation=0MrJm1ZTfW>
- Pixlr. (s.f.). Generador de imágenes – Pixlr. Recuperado 14 mayo 2025, desde <https://pixlr.com/es/image-generator/>
- PHP.net. (s.f.). array_merge – Manual de PHP. Recuperado 14 mayo 2025, desde <https://www.php.net/manual/en/function.array-merge.php>
- Stack Overflow. (s.f.). React JS: setInnerHTML vs dangerouslySetInnerHTML. Recuperado 14 mayo 2025, desde <https://stackoverflow.com/questions/37337289/react-js-set-innerhtml-vs-dangerouslysetinnerhtml>
- GeeksforGeeks. (s.f.). How to place cursor position at end of text in text input field using JavaScript. Recuperado 14 mayo 2025, desde <https://www.geeksforgeeks.org/how-to-place-cursor-position-at-end-of-text-in-text-input-field-using-javascript/>
- Programación con Dani. (2023, 18 de septiembre). Symfony 6 Mailer: cómo enviar emails y plantillas [Video]. YouTube. <https://www.youtube.com/watch?v=ECEwmZYV8P4&t=901s>
- Symfony. (s.f.). Mailer. Recuperado el 14 de mayo de 2025, de <https://symfony.com/doc/current/mailer.html>

7. ANEXOS

ANEXOS GENERALES SOBRE LA APLICACIÓN

1. ANEXO 1. GUÍA DE ESTILOS

Introducción

El objetivo principal de esta guía de estilos es crear un marco organizado para el diseño y desarrollo de la interfaz de usuario en mi aplicación GUEXT1080 proporcionada por la empresa. También, es importante garantizar una estructura coherente y un diseño lo más parecido en todas las secciones que haya de la aplicación, asegurando que la creación de los contenidos sea fácil y visualmente atractiva.

Mi propósito es integrar un editor WYSIWYG llamado Summernote para la creación de plantillas de correos electrónicos, verificando una experiencia intuitiva y eficiente. En estas plantillas, los usuarios tendrán la posibilidad de insertar imágenes, placeholders de variables y gestionar múltiples versiones en diferentes idiomas ya sea inglés, español, etc. Gracias a la interfaz, facilitará la incorporación de estos elementos de forma visual y accesible, mejorando la usabilidad del sistema.

Asimismo, esta guía está compuesta por elementos como tipografía, colores, imágenes, iconografía y una estructura de maquetación web, que aseguran una presentación clara y organizada del contenido. Además, esta sección que constará en el proyecto debe mantener una apariencia coherente y visualmente atractiva, teniendo en cuenta la usabilidad y accesibilidad para que los usuarios puedan gestionar las plantillas sencillamente. De este modo, la navegación será intuitiva permitiendo una interacción fluida con los elementos y garantizando una experiencia de uso eficaz y atractivo.

Una vez establecido esto, se busca optimizar la navegación del usuario, garantizando que cada elemento del diseño se ajuste a los requisitos definidos y que estas implementaciones propuestas sean de gran ayuda para los clientes.

Fotos y logos

Para las imágenes o fotografías han sido mejoradas con jpg o png y se mostrará las imágenes con una calidad profesional. En mi ocasión, no se enseñarán imágenes debido a la privacidad de datos y a la confidencialidad de la empresa.

Para mi proyecto, se dispondrá de algún ejemplo de imagen que es:



Tipografía

Ahora tenemos el apartado de tipografía, un elemento fundamental en el diseño web porque además de dar información, permite describir la personalidad visual de un sitio. Un documento aparte del texto cuenta con un encabezado y títulos. En nuestra web, habrá diferentes etiquetas que son `<h1>` y `<h2>` según lo importante que sea el texto y se pondrá para la web para una utilización u otra.

- **Fuente principal**

La tipografía que se empleará para toda la aplicación será **Roboto**, un formato de fuente Sans-Serif que permite una visibilidad adecuada en todos los tamaños y de esta forma, los usuarios no tendrán ningún tipo de problema a la hora de leer la información.

- **Uso (tamaño y color)**

Etiqueta H1:

Se utilizará para los títulos de cada sección de la página de manera que los usuarios puedan distinguir el contenido de cada una. Tendrá un tamaño de 40 puntos y el color de la letra será negro.

Etiqueta H2:

Esta etiqueta se aplicará para el encabezado de la página, junto al logo que estará en la parte izquierda y sirve para saber en qué consiste la aplicación. También, se le pondrá en color negro y con un tamaño de 35 píxeles.

Etiqueta H3:

El titular H3 tiene como objetivo mostrar el título principal que pertenece al asunto, al previsualizar la plantilla finalizada junto con sus valores reemplazados, dejando al usuario un modelo de como quedaría finalmente. Además, se colocará de color azul RAF (#18787F) con un tamaño de 28 píxeles.

Texto o párrafos:

El texto del cuerpo al igual que en las etiquetas restantes estará en **Roboto**. Se usará tanto para el contenido de la página como el pie de página. El tamaño será de 16 puntos, útil para proporcionar una lectura cómoda y fácil de seguir. Además, el color de la fuente será negro (#000000), garantizando un contraste apropiado para que el contenido se pueda visualizar de una manera más clara.

Botones

En cuanto a los botones de mi proyecto sobre editor de plantillas, se colocará el mismo tipo de letra. El tamaño del texto será de 16 píxeles, asegurando que sea fácil de leer de cara al usuario.

El color del texto se escogerá según el fondo del botón para lograr un contraste adecuado. Por ejemplo, si el botón es de color azul RAF (#18787F), el texto será con un color menta (#C8F3E5), para asegurar que sea lo más visible posible. También, la utilización del color azul RAF en los botones permanece la coherencia con otros elementos visuales del sitio, como los encabezados y otros elementos gráficos, estableciendo una armonía visual en el diseño.

- **Estilo o tipo de fuente**

En el proyecto se empleará estos estilos o tipos de fuente:

- **Negrita (bold):** Su propósito es remarcar la información o contenido principal y de esta forma, captar la atención de los usuarios. Será usado para los títulos de secciones y subtítulos, así como algunos textos que hay en la página y botones de la aplicación web como "Crear", "Actualizar/Guardar Nueva Plantilla", entre otros.

Colores y uso tipográfico de color

Los colores principales que se insertarán de cara al proyecto son:

Azul oscuro Seguridad, estabilidad, seriedad...	C:100% M:99% Y:44% K:60% R:24 G:18 B:51 HTML:#181233
Menta eléctrica Intuición, lealtad, tranquilidad...	C:62% M:0% Y:47% K:0% R:70 G:217 B:170 HTML:#46d9aa
Gris claro Neutralidad, elegancia, templanza...	C:12% M:8% Y:9% K:0% R:230 G:230 B:230 HTML:#e6e6e6
Azul Raf Transparencia, armonía, confianza...	C:82% M:31% Y:43% K:17% R:24 G:120 B:127 HTML:#18787f

En esta parte, se explicarán los colores que se aplicarán en el diseño de las funcionalidades propuestas para la aplicación de la empresa incluyendo sus códigos hexadecimales y valores en el modelo RGB para el código.

Primero, el texto será de color negro metiendo las notas al pie de página. A través de esta opción, mejorará la accesibilidad y facilitará la lectura, ya que un fondo claro con un texto oscuro asegurará que podamos leer la información de forma sencilla y rápida, disminuyendo la fatiga visual.

Para el fondo de la aplicación se aplicará un color gris claro (#e6e6e6) que permite leer el contenido al proporcionar un contraste equilibrado con el texto oscuro y otros elementos gráficos. Además de esto, nos transmite varias sensaciones como neutralidad, elegancia y templanza.

Gracias a este color, mejora la experiencia de usuario de forma cómoda dejando que los elementos más importantes de la interfaz se destaquen de manera efectiva.

El color azul oscuro (#181233) se empleará para el logo de la aplicación transmitiendo una sensación de seguridad, estabilidad y seriedad. Mediante este color, da una estética moderna garantizando que haya buena visibilidad en diferentes fondos.

En cuanto a los elementos principales del proyecto como los botones de eventos para realizar diversas acciones, se utilizará el color azul RAF (#18787F) ya que proporciona transparencia, relajación y armonía. Además, en otras ocasiones, nos puede generar una sensación de confianza.

Con respecto al efecto hover, los colores del botón se alteran de manera que el fondo será con el color menta claro (#C8F3E5) y el texto con azul RAF (#18787F).

















Asimismo, el color verde magenta o menta eléctrica (#46d9aa) que será implementado para las opciones del menú del sistema y así, será más fácil para el usuario identificar con claridad las secciones de la web.

El objetivo principal de este color es que el usuario sepa en que opción del menú se encuentra en esos momentos. También, nos transmite una sensación de intuición y tranquilidad.

Iconografía

Para este punto, se incorporarán iconos que proporcionen operatividad dentro del proyecto, facilitando la creación de plantillas de email. Su objetivo es dar una navegación intuitiva, permitiendo a los usuarios identificar cada una de las acciones de la web y secciones del editor sin necesidad de ampliar la hoja para ubicarlas.

Los iconos que habrá en el proyecto de GUEXT 1080 son:

 Ver plantilla	 Contraseña
 Editar	 Salir / Cerrar sesión
 Eliminar	 Asunto
 Creación	 Contexto
 Ayuda	 Contenido
 Volver	 Actualizar plantilla
 Recargar	 Prueba Envío
 Enviar	 Limpiar

Respecto a esta parte, vemos una serie de iconos que se añadirán en la aplicación proporcionados de la librería PrimeReact como el icono de la basura en caso de que deseáramos eliminar una plantilla o el del ojo para visualizar la información de la plantilla que hemos seleccionado en ese momento.

Además de lo comentado, están diseñados para mejorar la navegación, proporcionando una experiencia más cómoda y viable para el usuario, pues su propósito es claro, ayuda a que sea una interfaz bien organizada y accesible en el que los usuarios puedan determinar las acciones de manera rápida y eficaz.

Estructura: Maquetación web

Descripción de la estructura:

Horizontalidad

La aplicación web GUEXT1080 está estructurada en una disposición horizontal, organizando sus secciones y elementos principales en una sola fila o en varias líneas alineadas de manera horizontal. Con la ayuda de este diseño, los usuarios podrán desplazarse sin problema entre las diferentes áreas o secciones del editor, lo que optimiza considerablemente la navegación.

Además de optimizar la accesibilidad, ofrece una experiencia de usuario más clara. De este modo, tanto los usuarios que usan con frecuencia la aplicación como aquellos que entran por primera vez pueden encontrar con rapidez las herramientas y secciones necesarias, permitiendo una interacción apropiada.

Jerarquización de la Información

La jerarquía visual remarca los aspectos fundamentales y los principios que se utilizan usualmente en el diseño web. Hay algunas maneras de organizar la información, pero todas ayudan a aumentar la efectividad de la comunicación.

La lógica visual se basa en la claridad y la simplicidad, donde todos los elementos tienen que estar conectados. A través de un diseño gráfico más sencillo, se pueden destacar las opciones más esenciales para el usuario.

Espacio de Trabajo

El área del contenido principal es la zona más importante de la página, donde se presenta la información considerable y donde los usuarios pueden entrar para visualizar los detalles más imprescindibles.

Respecto al proyecto, estos contenidos estarán ubicados en la parte central del editor. La información se organiza de forma clara a través de elementos visuales como menús desplegables y botones interactivos, lo que facilitará la navegación y hará la navegación más atractiva para el usuario.

Asimismo, el tamaño y tipo de fuente será más sencillo para la búsqueda de información. Un ejemplo sería la tabla de listado de las plantillas donde se muestra toda la información necesaria con textos o párrafos cortos de una manera destacada y así, la navegación será más comprensible llamando la atención de los usuarios.

Bloques de Estructura

La estructura estará compuesta mediante etiquetas <DIV>, que agrupan el contenido principal que estará en el cuerpo del texto. Estas etiquetas dejan aplicar estilos CSS personalizados, lo que permite editar cambios con respecto a los colores, el posicionamiento de las cajas, la tipografía y otras propiedades de diseño.

También, se tiene en cuenta el diseño responsive, permitiendo que la interfaz se adapte y se pueda visualizar en diferentes dispositivos como móviles, tablets, etc. A través de esto, proporciona una experiencia apropiada en cualquier dispositivo de manera fluida para todo tipo de usuarios.

Estudio de los Elementos del sitio web:

Para la aplicación de gestión de plantillas, he implementado varios elementos que estarán agrupados en base a las cuatro categorías requeridas. A continuación, se clasificarán de la siguiente forma:

- **Elementos de identificación:** Facilitan al usuario reconocer a quien corresponde la aplicación web. Algunos ejemplos de estos elementos son la cabecera junto con el logotipo ubicado en la parte superior izquierda de la página, el nombre ficticio del hotel de la empresa y la lista de opciones que hay a la derecha como “Ayuda”, “Contraseña” y “Cerrar Sesión”.
- **Elementos de navegación:** La función principal es facilitar el acceso a las distintas secciones de la aplicación y volver a la página principal cuando sea necesario. En este caso, los elementos de navegación son el menú de opciones que proporciona enlaces directos como "Ayuda", "Contraseña" y "Cerrar Sesión".
- **Elementos de interacción:** Estos tipos de elementos se utilizan para que los usuarios puedan interactuar y realizar acciones dentro de la interfaz del proyecto web. Algunos ejemplos son los menús desplegables disponibles en la pantalla de edición de plantillas, que permiten seleccionar el idioma, contexto y la plantilla deseada.

También, se incluyen botones como “Actualizar Plantilla”, así como los que hay con iconos en la tabla de listado de plantillas, que dejan visualizar la plantilla seleccionada, acceder a su edición o eliminarla. A través de esto, nos posibilita la interacción que tiene el usuario con el sistema directamente, aumentando la experiencia de uso y que sea eficiente.

- **Elementos de Contenido:** Tienen como finalidad mostrar la información principal que el usuario necesita dentro de la aplicación. Un ejemplo de ello es la tabla de plantillas que se compone de datos como nombre, asunto, contexto, contenido del mensaje y las acciones disponibles.

Asimismo, el contenido que aparece en el editor HTML de Summernote para el envío de correos electrónicos, que se visualiza al pulsar una plantilla elegida. Además, se utiliza un contenedor tipo “Dialog” para observar la información detallada de una plantilla cuando le damos en el icono del ojo, dejando entrar a la información de manera completa, organizada y fácil de leer.

Descripción de la estructura de la página de listado de plantillas:

El proyecto está formado en tres secciones primordiales como la parte superior, el contenido central y el pie de página.

El objetivo de la **parte superior de la página** es que los usuarios puedan comprender en qué sección están navegando actualmente. En este aspecto, visualizamos en la parte izquierda el logo de la empresa que he desarrollado seguido del nombre llamado “Demo Hotels”. En el lado derecho, se incluyen una lista con tres enlaces junto con sus iconos que son “Ayuda”, “Contraseña” y “Cerrar Sesión”.

La **parte central** de la página en GUEXT1080 describe el área principal de contenido, creado para insertar la información más significativa de una forma más precisa y organizada. En primer lugar, observamos un título en el que se destaca en que sección se encuentra el usuario que será “Listado de Plantillas” y así, tenga una orientación cuando interactúa por la aplicación. Después, tenemos un apartado de filtros de búsqueda para mostrar registros específicos de las plantillas en el que vemos que hay un menú desplegable para elegir un contexto que deseemos y un campo de texto para buscar por nombre, asunto o contenido de la plantilla. Además, debajo de estos elementos habrá un botón para limpiar el contenido que hayamos seleccionado.

A continuación, en la parte de abajo, disponemos de otro botón para crear una plantilla en el que si pulsamos podremos introducir un nombre que queramos y luego, la añadiremos. Luego, está la tabla que muestra la información de cada plantilla como **id**, **contexto**, **nombre**, **asunto**, **contenido** y **acciones**. En la columna “Acciones”, hay iconos que si pulsamos en ellos podremos visualizar, editar o eliminar ese registro. Posteriormente, tenemos un sistema de paginación en la parte inferior que ayuda en la navegación pasando de una página u otra y también, una opción para mostrar el número de plantillas que deseamos.

En la sección inferior, se situará el pie de página o “footer”, que servirá para colocar la información de contacto de la empresa tanto en el lado izquierdo como el derecho.

Descripción ventana emergente con texto e iconos (Información plantilla):

En esta pantalla, una vez pulsamos en el botón con el icono del ojo dentro de una plantilla, accederemos a una ventana emergente o modal donde se mostrará toda la información sobre la plantilla que el usuario haya escogido.

Esta ventana emergente aparece centrada en la pantalla, colocada sobre la interfaz principal y contiene todos los datos destacados de la plantilla y así, el usuario pueda consultarlos sin tener que salirse de ahí. Dentro del modal, se presenta el contenido que en primer lugar vemos que está el título que en este caso será el nombre de la plantilla. Luego debajo se resalta su contexto, más abajo tenemos el asunto y el contenido para que podamos revisar todo lo que se haya escrito.

Finalmente, dispone de un botón de acción que es “Salir” que deja al usuario salir del modal y regresar a la tabla sobre el listado de plantillas. Mediante este componente, aumenta la experiencia del usuario porque tenemos la posibilidad de ver todo lo relacionado con la plantilla de una forma sencilla y cómoda.

A continuación, en la siguiente hoja se observa una imagen sobre cómo se ha realizado la estructura de esta pantalla con su contenido.

Demo Hotels

Ayuda

Contraseña

Cerrar sesión

Listado de Plantillas

Filtros

Contextos

Seleccionar

+ Crear

		Contenido			Acciones
1	USER_MANAGEMENT	USER_RESET_PASSWORD	GUEST 1080: Restablecimiento de con...	Este es un mensaje de restablecimiento de contraseña	<div></div>
2	SHOP_RESERVATION_MAN...	CASE_ALERT	GUEST 1080: Alerta sobre tarea	Este es un mensaje de una alerta sobre tarea en el qu...	<div></div>
3	SECURITY_MANAGEMENT	GUEST_NAME	GUEST 1080: Seguridad de la habitación	Contenido de la seguridad de la habitación de un hotel	<div></div>

<

1

2

3

4

5

>

CASE_ALERT

Contexto: SHOP_RESERVATION_MANAGEMENT

Asunto: GUEST 1080: Alerta sobre tarea

Contenido: Esto es un mensaje de una alerta sobre tarea en el que se resume tu tarea

Salir

Powered By Nethits

© Version 1.43

Páginas interiores de texto y texto con editor (Sección Edición Plantillas)

La estructura principal es muy similar al de la página del listado de plantillas, pues tenemos la misma cabecera superior con sus enlaces y su pie de página. No obstante, cambia el cuerpo central que tiene una distribución horizontal acompañado de su contenido. Una vez pulsamos en el botón de editar que hay en la página del listado de plantillas, accedemos a la sección de “Edición Plantillas” que se comentará a continuación.

En primer lugar, se ha creado un contenedor de manera que quede mejor con el fondo y dentro tenemos un título de tipo h1 que servirá para saber en qué pantalla nos encontramos. Más abajo, veremos que hay una barra horizontal con varios menús desplegables que son para escoger el idioma, el contexto, luego un menú con varias opciones de plantillas que varían según el contexto que hayamos elegido. Además, variables en el que los usuarios podrán añadirlas una vez esté introducida la plantilla dentro del editor. Al lado de estos desplegables, hay un botón para borrar tanto el contenido que hay en el editor como el nombre de la plantilla que hayamos escrito.

Debajo de este componente, se presenta un campo de texto para colocar o modificar el nombre de la plantilla. Seguidamente de esto, está el editor de plantillas de email a través del plugin **Summernote** en el que contiene varias opciones de colores, tamaños, tipo de letra, etc.

Mediante esta herramienta, los usuarios tendrán la posibilidad de editar o escribir nuevas plantillas a partir de las que hay en las opciones, pues los usuarios escogerán alguna de las opciones de las plantillas y una vez hayamos pulsado en ella, aparecerá el contenido en el editor de plantillas automáticamente y podremos modificarla como mejor nos convenga.

En la parte inferior del área de contenido, hay un botón que es **“Actualizar Plantilla”** para actualizar una plantilla, pues aparecerá esta opción en el caso en el que solamente queramos editar su contenido sin cambiar su nombre o **“Guardar Nueva Plantilla”** que se guardaría como nueva si el usuario desea modificarlo todo.

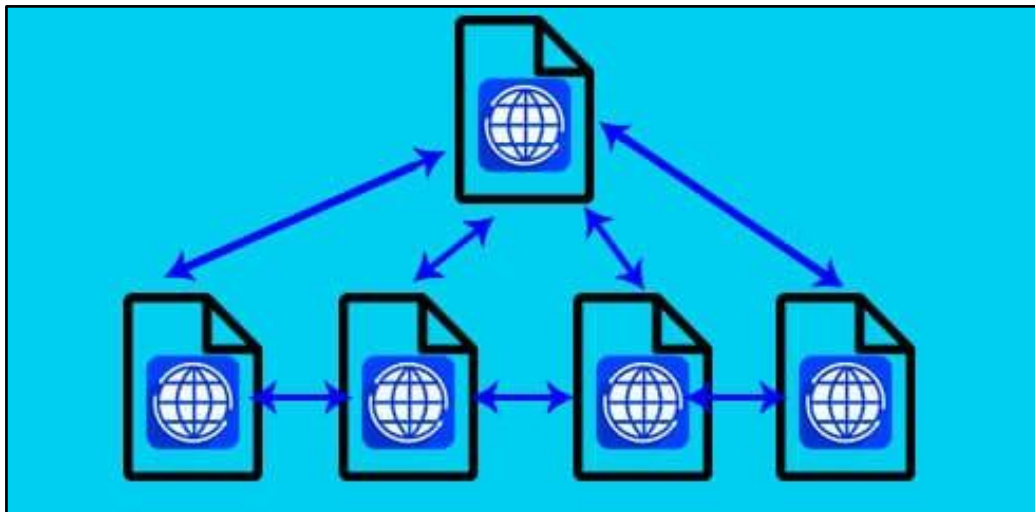
Finalmente, está el pie de página que es igual al de la anterior página como se ha comentado anteriormente. Y como comprobamos, tenemos una imagen sobre la estructura de esta sección.

The screenshot shows a web application interface titled "Edición Plantillas" (Template Editing) within a "Demo Hotels" context. The interface includes a header with a logo, the application name, and navigation links for "Ayuda", "Contraseña", and "Cerrar sesión". The main content area features a title "EDICIÓN PLANTILLAS" and a toolbar with dropdown menus for "Idioma", "Contextos", "Plantillas", and "Variables", along with a red trash icon. Below the toolbar is a form labeled "Nombre Plantilla:" with a placeholder "Introduce nombre de plantilla". A rich text editor follows, with a toolbar containing bold, italic, underline, style, list, and font size options. The editor area is currently empty. At the bottom of the editor is a blue button labeled "Actualizar/Guardar Nueva Plantilla". The footer of the application displays "Powered By Nethits" and "© Version 1.43".

Estructura: Mapa de navegación

El tipo de estructura de navegación que se emplea en esta aplicación en la empresa es lineal jerárquica, ya que todas las páginas están conectadas entre sí y es utilizada cuando tienes un volumen de información alto.

Aparte de eso, es el tipo más recomendado en proyectos complejos porque posibilita al usuario navegar entre las diferentes secciones y subsecciones de forma lógica y ordenada proporcionando una mejor experiencia al usuario. Otra ventaja principal es que facilita la escalabilidad, pudiendo añadir nuevas secciones sin afectar la navegación, lo que asegura una experiencia coherente cuando el contenido aumenta.



LINEAL JERÁRQUICA