



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una aplicación web para el control y
simulación de la producción de coches en Ford

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Bennor Haliba, Reda

Tutor/a: Valderas Aranda, Pedro José

Cotutor/a externo: TORRES QUILIS, JAVIER

CURSO ACADÉMICO: 2022/2023

Resum

L'objectiu d'aquest projecte és crear una eina que permeti als responsables de la fàbrica monitoritzar i optimitzar el procés de producció de cotxes. L'aplicació web mostrarà informació a temps real sobre l'estat de la fàbrica, els recursos disponibles, la producció pendent i els possibles problemes que puguin sorgir. A més, permetrà a l'usuari interactuar amb el simulador de la fàbrica, que recrearà el seu funcionament sota diferents condicions i escenaris. Per facilitar la comunicació entre l'aplicació web i la base de dades, es desenvoluparà una api que farà d'intermediari entre ambdues parts. L'api s'encarregarà de rebre les peticions de l'aplicació web i enviar les respostes amb les dades sol·licitades.

Paraules clau: web, monitorització, optimització, producció, cotxes, ford

Resumen

El objetivo de este proyecto es crear una herramienta que permita a los responsables de la fábrica monitorizar y optimizar el proceso de producción de coches. La aplicación web mostrará información en tiempo real sobre el estado de la fábrica, los recursos disponibles, la producción pendiente y los posibles problemas que puedan surgir. Además, permitirá al usuario interactuar con el simulador de la fábrica, que recreará el funcionamiento de la misma bajo diferentes condiciones y escenarios. Para facilitar la comunicación entre la aplicación web y la base de datos, se desarrollará una api que hará de intermediario entre ambas partes. La api se encargará de recibir las peticiones de la aplicación web y enviar las respuestas con los datos solicitados.

Palabras clave: web, monitorizar, optimizar, producción, coches, ford

Abstract

The aim of this project is to create a tool that allows the factory managers to monitor and optimise the car production process. The web application will display real-time information of the status of the factory, available and pending resources, and possible problems that may arise. It will also allow the user to interact with the factory simulator, which will recreate the operation of the factory under different conditions and scenarios. To facilitate communication between the web application and the database, an api will be developed to act as an intermediary between both parties. The api will be in charge of receiving the requests from the web application and sending back the responses with the requested data.

Key words: web, monitoring, optimise, production, cars, ford

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VII
<hr/>	
1 Introducción	1
1.1 Contexto y Motivación	1
1.2 Objetivos	1
1.3 Estructura de la Memoria	2
2 Contexto empresarial	4
2.1 Historia y Fundación	4
2.2 Hitos Importantes	4
2.3 Contribuciones Significativas	5
2.4 Situación tecnológica actual	5
2.5 Necesidades	6
3 Metodología	8
3.1 Requisitos y Planificación	8
3.2 Creación de Bocetos y Prototipos	8
3.3 Desarrollo Iterativo	9
3.4 Retroalimentación y Ajustes	10
3.5 Adición de Nuevas Páginas y Funcionalidades	10
4 Análisis de requisitos	11
4.1 Casos de Uso	11
4.2 Escenarios	12
5 Solución Desarrollada	13
5.1 Visión General de la Solución	13
5.2 Elementos de la Solución	14
5.2.1 Dashboard Principal	14
5.2.2 Simulador de Producción	15
5.2.3 Predictor	15
5.2.4 Exit Paint	15
5.2.5 Knowlex (Esquema de Planta)	15
5.2.6 Schedules y Control Room	16
5.2.7 Gestión de Tareas y Comunicación Interna	16
6 Diseño de la API REST	17
6.1 Introducción	17
6.2 Tecnologías Utilizadas	17
6.3 Estructura de la API	17
6.4 Métodos HTTP	18
6.5 Respuestas JSON	18
6.6 Ejemplo de Endpoint y Respuesta	18
7 Diseño y Desarrollo del Dashboard	20
7.1 Tecnologías y Bibliotecas Utilizadas	20

7.1.1	Bootstrap Icons	20
7.1.2	jQuery Slim	20
7.1.3	Popper.js	20
7.1.4	Bootstrap	20
7.1.5	Dashboard Stylesheet	21
7.1.6	GoJS	21
7.1.7	Moment.js	21
7.1.8	Highcharts	21
7.1.9	Highcharts Exporting y Highcharts Export Data	21
7.1.10	Highcharts Accessibility	21
7.2	Diseño del Dashboard	21
7.3	Componentes Destacados del Dashboard	22
7.3.1	Gráficos Interactivos	22
7.3.2	Diagramas Dinámicos	22
7.3.3	Filtros y Controles	22
7.3.4	Exportación de Datos	22
7.4	Bocetos de las Interfaces de la Aplicación	22
7.5	Accesibilidad y Responsividad	23
7.6	Conclusión	23
8	Desarrollo de la Solución	27
8.1	Arquitectura del Sistema	27
8.1.1	Componentes de la Arquitectura	27
8.1.2	Comunicación en Tiempo Real	28
8.1.3	Interacción y Flujo del Usuario	28
8.1.4	Seguridad y Privacidad	28
8.1.5	Escalabilidad y Mejoras Futuras	28
8.1.6	Conclusiones	29
8.2	Ejemplos de Código	29
8.2.1	Ejemplos de Endpoints y Funciones	29
8.3	Ejemplos de la pagina de Tareas	31
8.3.1	Código HTML, CSS y JavaScript del Panel de Control de Tareas	31
8.3.2	Ejemplos Representativos de Otras Partes del Sistema	33
8.3.3	Estructura del JSON de Usuarios	35
8.3.4	Ejemplo de JSON de Tareas	35
9	Producto desarrollado	38
9.1	Presentación de la Aplicación	38
9.1.1	Interfaz de Inicio de Sesión	38
9.1.2	Panel Principal	39
9.2	Escenarios de Uso	39
9.2.1	Monitorización de Producción	39
9.2.2	Simulación de Producción	40
9.2.3	Predicción de Tiempos de Paso por Zona	42
9.2.4	Análisis de Programas de Producción (Schedules)	44
9.2.5	Creación y Seguimiento de Tareas	45
10	Validación	49
10.0.1	Análisis de la Página Principal	50
10.0.2	Conclusiones	50
11	Conclusions	51
	Bibliografía	52

Índice de figuras

4.1	Diagrama de casos de uso"	12
5.1	Arquitectura General de la Solución	13
7.1	Boceto del Dashboard	24
7.2	Boceto del Simulador	24
7.3	Boceto del Predictor	25
7.4	Boceto de Schedules	25
7.5	Boceto de Control Room	26
7.6	Boceto de Gestión de Tareas	26
9.1	Pantalla de Inicio de Sesión	38
9.2	Panel Principal	39
9.3	Panel Principal	39
9.4	Panel Principal	40
9.5	Panel de Simulación	41
9.6	Esquema de la Fábrica	41
9.7	Configuración de la Simulación	42
9.8	Ejecución de la Simulación	43
9.9	Panel Principal	44
9.10	Página de Predicción	44
9.11	Panel Principal	45
9.12	Schedules detallados	45
9.13	Panel Principal	46
9.14	Panel Principal	47
9.15	Detalles Tarea	47

Índice de tablas

3.1	Comparación Metodologías	9
-----	--------------------------	---

CAPÍTULO 1

Introducción

1.1 Contexto y Motivación

En el competitivo entorno actual de la industria automotriz, Ford Motors se esfuerza por mantenerse a la vanguardia al mejorar constantemente sus procesos de producción. La eficiencia operativa, la calidad del producto y la satisfacción del cliente son aspectos cruciales para el éxito de Ford. En este contexto, la monitorización y la optimización del proceso de producción de coches se han vuelto esenciales para garantizar resultados superiores y mantener su posición de liderazgo en el mercado.

La motivación detrás de este proyecto de Trabajo Fin de Grado (TFG) radica en la necesidad de desarrollar una herramienta web personalizada para Ford Motors. Esta herramienta permitirá a los responsables de la fábrica de coches monitorizar y optimizar de manera efectiva el proceso de producción, mejorando la eficiencia, la planificación y la toma de decisiones.

1.2 Objetivos

El objetivo principal de este proyecto de TFG es desarrollar una herramienta web integral y personalizada para Ford Motors que abarque los siguientes aspectos:

1. Monitorización en tiempo real: La herramienta proporcionará un dashboard con una representación visual del esquema de la fábrica, permitiendo a los responsables detectar paradas o interrupciones en el proceso de producción de forma rápida y eficiente. Además, se mostrarán tablas y gráficos actuales y históricos de la disponibilidad de patines, recursos clave en la producción de coches.
2. Simulación y optimización: La herramienta contará con un simulador interno que permitirá recrear y evaluar diferentes escenarios y condiciones de producción. Esto facilitará la planificación estratégica y la toma de decisiones informadas para optimizar los recursos y maximizar la eficiencia del proceso.
3. Predicción del tiempo de producción: Se implementará un predictor impulsado por inteligencia artificial (IA) que, dado un número de identificación de vehículo (VIN), estimará el tiempo requerido para que cada zona de la fábrica complete su proceso de producción. Esta funcionalidad permitirá una planificación más precisa y una asignación adecuada de los recursos.
4. Seguimiento de schedules y control de ganancias/pérdidas: La herramienta incluirá un apartado dedicado a visualizar los programas de producción actuales (sche-

dules), proporcionando una visión actualizada de las actividades planificadas en la fábrica. Además, se implementará un control room en tiempo real que mostrará un balance de ganancias o pérdidas en relación con los objetivos de producción establecidos. Esto permitirá a los responsables identificar desviaciones y tomar medidas correctivas de manera oportuna.

5. **Gestión de tareas y comunicación interna:** Se incluirá un apartado de tareas donde los coordinadores y supervisores podrán asignar tareas específicas a los analistas. Este apartado también contará con un chat privado, lo que facilitará la comunicación interna y agilizará la resolución de problemas.

En resumen, este proyecto de TFG tiene como objetivo desarrollar una herramienta web personalizada para Ford Motors, que permita monitorizar y optimizar el proceso de producción de coches, mejorar la toma de decisiones y la planificación, así como facilitar la comunicación interna y la gestión eficiente de tareas.

1.3 Estructura de la Memoria

Esta memoria está organizada de la siguiente manera para proporcionar una visión completa y ordenada del trabajo realizado:

1. **Introducción:** En este capítulo se introduce el contexto y la motivación del proyecto, se establecen los objetivos y se describe la estructura general de la memoria.
2. **Contexto empresarial:** En este capítulo, se presenta el contexto de la empresa en la que se desarrolló el proyecto. Se describe la historia y fundación de la empresa, los hitos importantes en su desarrollo y las contribuciones significativas al campo. Además, se examina la situación tecnológica actual de la empresa y se destacan las necesidades que condujeron al proyecto.
3. **Metodología:** Este capítulo detalla la metodología seguida durante todo el proceso de desarrollo. Se explican los pasos desde la definición de requisitos hasta la implementación, incluyendo la creación de bocetos y prototipos, el desarrollo iterativo y la retroalimentación recibida.
4. **Análisis de requisitos:** Aquí se presentan los casos de uso y escenarios que sirvieron como base para el diseño y desarrollo del sistema. Se detallan las funcionalidades clave que se esperaban y cómo se tradujeron en requisitos específicos.
5. **Solución Desarrollada:** En este capítulo, se ofrece una visión general de la solución desarrollada. Se describen los elementos que componen la aplicación, desde el dashboard principal hasta las herramientas específicas como el simulador de producción, el predictor y otros. Se destaca cómo cada componente contribuye a la funcionalidad general del sistema.
6. **Diseño de la API REST:** Aquí se introduce la API REST desarrollada como parte del proyecto. Se describen las tecnologías utilizadas, la estructura de la API, los métodos HTTP utilizados y cómo se manejan las respuestas JSON. Se proporciona un ejemplo de un endpoint y su respuesta.
7. **Diseño y Desarrollo del Dashboard:** Este capítulo se centra en el diseño y desarrollo del dashboard de la aplicación, que es una parte central de la interfaz de usuario. Se detallan las tecnologías y bibliotecas utilizadas, se explican los componentes clave del dashboard y se aborda la accesibilidad y la responsividad.

8. **Bocetos de las Interfaces de la Aplicación:** Se presentan los bocetos iniciales de las interfaces de la aplicación, mostrando cómo evolucionó el diseño desde las primeras ideas hasta la versión final.
9. **Desarrollo de la Solución:** En este capítulo se profundiza en la arquitectura del sistema, con detalles sobre la comunicación en tiempo real, la interacción del usuario, la seguridad y la privacidad. Se incluyen ejemplos de código representativos y se presenta la estructura de los datos en formato JSON utilizada en la aplicación.
10. **Producto desarrollado:** Este capítulo presenta el producto final desarrollado, incluyendo la interfaz de inicio de sesión y el panel principal. Se describen los escenarios de uso clave que ilustran cómo los usuarios interactúan con la aplicación.
11. **Validación:** Aquí se analiza la validación realizada, centrándose en la evaluación de la página principal y proporcionando conclusiones basadas en los resultados de la validación.
12. **Conclusiones:** Se presentan las conclusiones generales del proyecto y se destacan los logros y desafíos enfrentados durante el desarrollo. Se proporciona una visión general de las lecciones aprendidas y las posibles mejoras futuras.
13. **Bibliografía:** Se enumera la bibliografía y las fuentes consultadas durante el proyecto, incluyendo documentos, libros y recursos en línea.

Esta estructura ha sido diseñada para proporcionar una narrativa lógica y coherente de todo el proceso de desarrollo de la aplicación, desde su concepción hasta su implementación y validación.

CAPÍTULO 2

Contexto empresarial

En este capítulo, se analizará en profundidad el contexto empresarial de Ford Motor Company, destacando su historia, hitos importantes y contribuciones notables a la industria automotriz.

2.1 Historia y Fundación

Ford Motor Company, conocida mundialmente como Ford, es una empresa multinacional originaria de Estados Unidos y un actor influyente en la industria automotriz. La compañía fue fundada por el visionario industrial Henry Ford, quien junto con un grupo de inversores, estableció la empresa el 16 de junio de 1903 [4]. Desde sus inicios, Ford se ha centrado en la fabricación y venta de automóviles y vehículos comerciales bajo la marca Ford, así como en la producción de vehículos de lujo bajo la marca Lincoln [4].

2.2 Hitos Importantes

A lo largo de su extensa historia, Ford Motors ha logrado una serie de hitos significativos que han dejado una huella indeleble en la industria automotriz:

- **1903: Fundación de Ford Motor Company:** Con un capital inicial de 28,000 dólares aportados por once inversores, Henry Ford estableció la compañía, sentando las bases para una revolución en la movilidad y la fabricación de automóviles [4].
- **1913: Creación de Ford Motors Argentina:** Ford extendió su presencia internacional al inaugurar su segunda filial fuera de Norteamérica en Argentina. Esta expansión marcó el comienzo de una influencia global en la industria [2].
- **1916: Inauguración de la primera planta en la calle Herrera:** La primera planta de Ford en el barrio de Barracas, Argentina, comenzó sus operaciones, marcando un paso fundamental en la producción localizada de vehículos [2].
- **1918: Producción íntegra en Argentina:** Ford logró un hito al ensamblar vehículos íntegramente en Argentina a una tasa de 48 unidades diarias, consolidando su presencia y contribución a la economía del país [2].
- **1922: Modelo Ford T:** La planta de La Boca inició la producción del legendario modelo Ford T, que se convertiría en un símbolo icónico de la revolución automotriz que Ford lideró en la primera mitad del siglo XX [2].

2.3 Contribuciones Significativas

Las contribuciones de Ford Motors a la industria automotriz han sido profundas y transformadoras, gracias a la visión y la innovación de Henry Ford:

- **Visión Globalizadora:** Henry Ford tuvo una visión global de la industria del automóvil, lo que lo llevó a exportar un sistema de franquicias a nivel mundial. Esta estrategia de expansión contribuyó a la presencia global de Ford y a su impacto en la movilidad a nivel internacional [3].
- **Fordismo y Producción en Cadena:** La introducción del sistema de producción en cadena, conocido como "fordismo", revolucionó la industria automotriz y estableció un nuevo estándar en eficiencia y producción masiva. Este enfoque permitió a Ford fabricar grandes cantidades de automóviles a bajos costos, acelerando la disponibilidad de vehículos a un público más amplio [3].
- **Legado Duradero:** La influencia del "fordismo" y las innovaciones de producción de Henry Ford siguen resonando en la industria automotriz y más allá. El término "fordismo" se ha utilizado para describir sistemas de producción similares en diversas industrias, destacando la duradera influencia de Ford en los métodos de fabricación y operaciones [3].

Las contribuciones y logros de Ford Motors en su historia ilustran su impacto duradero en la industria automotriz, desde sus raíces en la producción pionera de automóviles hasta su influencia en la producción en cadena y la movilidad global.

2.4 Situación tecnológica actual

En el área de control de producción de Ford Motors, se han experimentado una serie de cambios significativos debido a la reestructuración de la producción, que incluye la reducción de modelos y un aumento en la cantidad de vehículos de los modelos existentes. Estos cambios en la estrategia de producción han generado la necesidad de probar nuevas estrategias y requerido herramientas más avanzadas de monitorización y simulación. A continuación, se detallan las principales deficiencias tecnológicas encontradas y la necesidad de soluciones adecuadas:

1. **Interfaz de usuario obsoleta:** Las herramientas utilizadas presentaban una interfaz de usuario obsoleta y poco intuitiva, lo que dificultaba la adaptación a los nuevos cambios en la producción. La visualización de la información era confusa y no permitía un análisis eficiente de los datos relevantes para la toma de decisiones estratégicas.
2. **Falta de actualización en tiempo real:** La falta de actualización en tiempo real en las aplicaciones y páginas web utilizadas dificultaba la comprensión de los cambios inmediatos en la producción. La capacidad de monitorizar en tiempo real los indicadores clave de rendimiento, como la producción por zona o la detección de paradas, se convirtió en un aspecto esencial para evaluar y ajustar rápidamente las estrategias de producción.
3. **Ausencia de simuladores:** La necesidad de probar nuevas estrategias de producción requería contar con herramientas de simulación avanzadas. La ausencia de

simuladores implicaba una mayor dificultad para evaluar los impactos de las decisiones en la producción y optimizar los recursos disponibles. Una herramienta de simulación adecuada permitiría a Ford Motors anticiparse a los desafíos y tomar decisiones informadas para maximizar la eficiencia y el rendimiento en el proceso de producción.

4. **Dificultad en la detección de paradas:** La detección de paradas en la producción se realizaba de manera manual y poco eficiente. La falta de herramientas automatizadas y actualizadas dificultaba la identificación temprana de paradas y afectaba la capacidad de respuesta rápida para solucionar los problemas. Se requería una solución de monitorización que permitiera detectar de manera precisa y oportuna cualquier interrupción en el proceso de producción.
5. **Dependencia de hojas de cálculo (Excel):** La dependencia excesiva de hojas de cálculo de Excel para realizar tareas y análisis específicos limitaba la escalabilidad y eficiencia de los procesos de producción. La necesidad de gestionar y actualizar múltiples hojas de cálculo dificultaba la centralización y compartición de datos, así como la automatización de procesos clave en la producción.

Estas limitaciones y desafíos tecnológicos evidencian la necesidad urgente de desarrollar una herramienta web personalizada que aborde las necesidades de monitorización y simulación en el nuevo contexto de producción de Ford Motors. Una solución adecuada proporcionará una interfaz intuitiva, actualizaciones en tiempo real, capacidades de simulación y monitorización efectivas para impulsar la eficiencia y el rendimiento en el proceso de producción.

2.5 Necesidades

En esta sección del Capítulo 2, se abordarán las necesidades específicas que surgieron a partir de las deficiencias tecnológicas identificadas en el contexto de producción de Ford Motors. Estas necesidades servirán como base para el desarrollo de la herramienta web personalizada. A continuación, se destacarán algunas de las principales necesidades identificadas:

1. **Interfaz intuitiva y moderna:** Existe una necesidad de contar con una interfaz de usuario intuitiva y moderna que facilite la visualización y el análisis de los datos de producción. La nueva herramienta web debe presentar un diseño atractivo y amigable para los usuarios, que les permita comprender fácilmente la información relevante y tomar decisiones informadas.
2. **Actualización en tiempo real:** Es fundamental contar con actualizaciones en tiempo real de los datos de producción. La herramienta web debe proporcionar información actualizada de forma instantánea, permitiendo a los responsables de la fábrica monitorear en tiempo real el estado de la producción, los recursos disponibles y cualquier problema que pueda surgir.
3. **Simulación de escenarios de producción:** La herramienta debe incluir capacidades de simulación que permitan probar diferentes escenarios de producción. Esto facilitará la toma de decisiones estratégicas al evaluar el impacto de posibles cambios en el proceso de producción y optimizar la asignación de recursos.
4. **Detección automática de paradas:** Se requiere una solución automatizada que permita la detección temprana y precisa de paradas en la producción. La herramienta

web debe ser capaz de identificar y notificar cualquier interrupción en el proceso de producción, lo que facilitará una respuesta rápida y eficiente para minimizar el impacto en la productividad.

5. **Gestión centralizada de datos:** Es esencial contar con una solución que centralice y facilite la gestión de los datos relacionados con la producción. La herramienta web debe permitir el almacenamiento y la visualización centralizados de datos, eliminando la dependencia de hojas de cálculo y mejorando la compartición y la colaboración entre los diferentes equipos.

Estas necesidades identificadas servirán como base para el diseño y desarrollo de la herramienta web personalizada, con el objetivo de mejorar la eficiencia y la efectividad del proceso de producción en Ford Motors.

CAPÍTULO 3

Metodología

En el desarrollo de la herramienta web para el control de producción en Ford Motors, se adoptó una metodología ágil y colaborativa, que permitió una respuesta efectiva a las necesidades cambiantes del departamento y aseguró la satisfacción de los interesados. A continuación, se describen las principales etapas de la metodología utilizada.

3.1 Requisitos y Planificación

La metodología comenzó con una fase de recopilación de requisitos, donde se llevó a cabo una reunión con el encargado del departamento (supervisor) para comprender completamente sus necesidades y expectativas. Se identificaron los problemas actuales del proceso de producción y se discutió cómo la herramienta web podía abordarlos de manera eficiente.

Posteriormente, se realizó una sesión de brainstorming con otros desarrolladores para discutir posibles tecnologías y enfoques para el desarrollo de la herramienta. Después de una investigación previa, se determinó que Python era una opción adecuada para desarrollar la API y, por lo tanto, se optó por desarrollar la herramienta web utilizando esta tecnología.

Con base en los requisitos recopilados y las tecnologías seleccionadas, se planificó el desarrollo de la herramienta en etapas incrementales, teniendo en cuenta la naturaleza cambiante del departamento y la posibilidad de agregar nuevas funcionalidades a medida que avanzaba el proyecto.

3.2 Creación de Bocetos y Prototipos

Una vez establecidos los requisitos y la planificación inicial, se procedió a crear bocetos a mano de la interfaz de la herramienta web. Estos bocetos se utilizaron para presentar la idea al supervisor y al coordinador, quien sería el principal usuario de la herramienta. La retroalimentación obtenida de ellos ayudó a refinar los diseños y garantizar que la interfaz fuera intuitiva y satisficiera las necesidades del departamento.

Con base en los bocetos aprobados, se desarrollaron prototipos de las primeras páginas de la herramienta. Estos prototipos permitieron visualizar cómo se vería y funcionaría la herramienta de manera preliminar, lo que facilitó la detección temprana de posibles mejoras y ajustes.

Metodología	Descripción	Ventajas	Desventajas
Iterativo	Un proceso repetitivo donde se implementan funcionalidades principales en las primeras etapas del proyecto [4]	Permite detectar fallos o ajustes necesarios en una etapa temprana [4]	Puede requerir más tiempo para completar el proyecto
Cascada	Un proceso lineal donde cada fase depende de la anterior [6]	Fácil de entender y gestionar	Difícil de realizar cambios una vez iniciado el proyecto
Ágil	Un enfoque flexible donde se realizan entregas frecuentes y se adaptan a los cambios rápidamente [5]	Permite adaptarse rápidamente a los cambios	Puede ser difícil predecir el tiempo y costo del proyecto
Prototipado	Un proceso donde se crea un prototipo inicial para obtener retroalimentación del cliente antes de desarrollar el producto final [6]	Permite obtener retroalimentación temprana del cliente	Puede requerir más tiempo y recursos para crear prototipos

Tabla 3.1: Comparación Metodologías

3.3 Desarrollo Iterativo

El desarrollo iterativo es un proceso de desarrollo de software creado en respuesta a las debilidades del modelo tradicional de cascada. Básicamente, este modelo de desarrollo, que no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones), es uno de los más utilizados en los últimos tiempos [4]. El modelo consta de diversas etapas de desarrollo en cada incremento, las cuales inician con el análisis y finalizan con la instauración y aprobación del sistema [4].

Se planifica un proyecto en distintos bloques temporales que se le denominan iteración. En una iteración se repite un determinado proceso de trabajo que brinda un resultado más completo para un producto final, de forma que quien lo utilice reciba beneficios de este proyecto de manera creciente [4]. Para llegar a lograr esto, cada requerimiento debe tener un completo desarrollo en una única iteración que debe de incluir pruebas y una documentación para que el equipo pueda cumplir con todos los objetivos que sean necesarios y esté listo para ser dado al cliente [4].

Existen numerosas metodologías de desarrollo de software, todas capaces de brindar los resultados que deseas. Algunas metodologías populares incluyen Desarrollo Ágil, Scrum, FDD: desarrollo basado en características [5]. La tabla 3.1 presenta una comparativa entre el desarrollo iterativo y otras metodologías populares:

Aunque existen varios enfoques de desarrollo de software, el enfoque iterativo fue elegido para este proyecto por varias razones:

1. **Naturaleza Evolutiva del Proyecto:** Dado que se estaba desarrollando una herramienta web compleja con múltiples funcionalidades, era probable que los requisitos cambiaran a medida que avanzaba el proyecto.

2. **Retroalimentación Continua del Supervisor:** La colaboración constante con el supervisor permitía ajustar y mejorar la herramienta en función de su evaluación y comentarios.
3. **Adaptación a las Necesidades del Usuario:** La naturaleza iterativa permitía adaptarse mejor a las necesidades reales de los usuarios, brindando un producto final más alineado con sus expectativas.
4. **Reducción de Riesgos:** El desarrollo iterativo ayudó a identificar problemas y desafíos en etapas tempranas, lo que redujo el riesgo de problemas mayores en el futuro.
5. **Flexibilidad para Agregar Funcionalidades:** La herramienta web debía adaptarse a las necesidades cambiantes de los usuarios y las demandas del supervisor, lo que hacía que el enfoque iterativo fuera más adecuado.

En resumen, el enfoque iterativo de desarrollo resultó ser el más apropiado para este proyecto debido a su capacidad de adaptación, entrega temprana y mejora continua, lo que permitió desarrollar una herramienta web funcional y eficaz que satisfizo las necesidades del usuario y las expectativas del supervisor.

3.4 Retroalimentación y Ajustes

La retroalimentación del supervisor fue esencial para mejorar y ajustar la herramienta web. Las reuniones regulares con el supervisor permitieron obtener comentarios sobre las funcionalidades existentes y discutir la incorporación de nuevas páginas o mejoras adicionales.

La colaboración cercana con el supervisor y el coordinador aseguró que la herramienta web cumpliera con los requisitos y necesidades específicas del departamento de control de producción.

3.5 Adición de Nuevas Páginas y Funcionalidades

Una vez que las páginas iniciales estaban en funcionamiento y cumplían con los requisitos, se procedió a agregar nuevas funcionalidades y páginas a la herramienta web. Estas adiciones se realizaron en respuesta a nuevas necesidades identificadas por el supervisor y otros miembros del departamento.

La metodología ágil y el enfoque iterativo permitieron una rápida incorporación de nuevas funcionalidades y ajustes en el desarrollo, lo que resultó en una herramienta web altamente personalizada y adaptada a las necesidades cambiantes del departamento de control de producción en Ford Motors.

En resumen, la metodología ágil, basada en la colaboración, la retroalimentación continua y el desarrollo iterativo, fue fundamental para el éxito del desarrollo de la herramienta web, asegurando que cumpliera con las necesidades del departamento y proporcionara una solución eficiente para el control de producción.

CAPÍTULO 4

Análisis de requisitos

En esta etapa del proyecto, se llevó a cabo un análisis exhaustivo de los requisitos para definir las funcionalidades y características clave que se implementarían en la herramienta web. El análisis se centró en entender las necesidades del departamento de control de producción en Ford Motors y en establecer los casos de uso, escenarios y diagramas de clases necesarios para el diseño y desarrollo efectivo de la herramienta.

4.1 Casos de Uso

Los casos de uso se utilizaron para representar las interacciones entre los diferentes actores y el sistema. Se identificaron los siguientes casos de uso principales:

1. **Monitorización de Producción:** Permite a los usuarios visualizar en tiempo real el estado de la producción en la fábrica, incluyendo la cantidad de coches producidos en diferentes zonas y la disponibilidad de patines para el transporte de los coches.
2. **Simulación de Producción:** Permite a los usuarios simular diferentes escenarios de producción y evaluar el rendimiento bajo distintas condiciones y programaciones. Esto ayuda en la toma de decisiones estratégicas para optimizar los recursos y mejorar la eficiencia del proceso de producción.
3. **Predicción de Tiempos de Producción:** Utilizando técnicas de inteligencia artificial, este caso de uso permite a los usuarios obtener estimaciones precisas sobre el tiempo que tomará completar cada zona de producción para un vehículo específico, lo que facilita la planificación y programación de la producción.
4. **Visualización de Schedules:** Proporciona a los usuarios la capacidad de ver los programas de producción actuales y futuros, lo que ayuda a coordinar y organizar el flujo de trabajo en la fábrica.
5. **Control Room en Tiempo Real:** Presenta un balance de ganancias o pérdidas en relación con los objetivos de producción establecidos, lo que permite a los responsables identificar desviaciones y tomar medidas correctivas en tiempo real.
6. **Gestión de Tareas y Chat:** Permite a los coordinadores y supervisores asignar tareas específicas a los analistas y mantener una comunicación eficiente a través de un chat privado.

4.2 Escenarios

Los escenarios se utilizaron para describir situaciones específicas en las cuales los usuarios interactúan con la herramienta web. Algunos ejemplos de escenarios incluyen:

1. **Monitorización de Producción:** Un supervisor utiliza la herramienta web para verificar el estado de la producción en tiempo real. Esto le permite identificar cualquier zona con producción detenida y tomar acciones inmediatas para resolver problemas. Por ejemplo, si una máquina se ha detenido, el supervisor puede enviar a un técnico para repararla y reanudar la producción lo antes posible.
2. **Simulación de Producción:** Después de un fallo en la zona de pinturas, el analista tiene que ajustar la cantidad de coches que puede producir con una cantidad menor proveniente de pinturas. El analista usa el simulador para probar varios ajustes y encontrar la configuración más adecuada. Esto le permite maximizar la producción a pesar del fallo en la zona de pinturas.
3. **Predicción de Tiempos de Producción:** Un coordinador ingresa el número de identificación de un vehículo en la herramienta web y obtiene estimaciones precisas sobre el tiempo requerido para completar cada etapa de producción. Esto le permite ajustar el cronograma de manera adecuada, asegurando que el vehículo se complete en el tiempo previsto.
4. **Visualización de Programas y Control Room en Tiempo Real:** Un supervisor revisa los programas de producción actuales y futuros para asegurarse de que estén alineados con los objetivos establecidos al mismo tiempo utiliza el control room para monitorear el progreso de la producción en relación con los objetivos. Si encuentra algún desajuste, puede realizar ajustes según sea necesario para asegurar que la producción se mantenga en línea con los objetivos.
5. **Gestión de Tareas y Chat:** Un coordinador asigna tareas específicas a los analistas y se comunica con ellos a través del chat privado para resolver problemas y garantizar una coordinación efectiva. Por ejemplo, si un analista necesita ayuda para resolver un problema, puede comunicarse con el coordinador a través del chat y recibir instrucciones claras sobre cómo proceder.

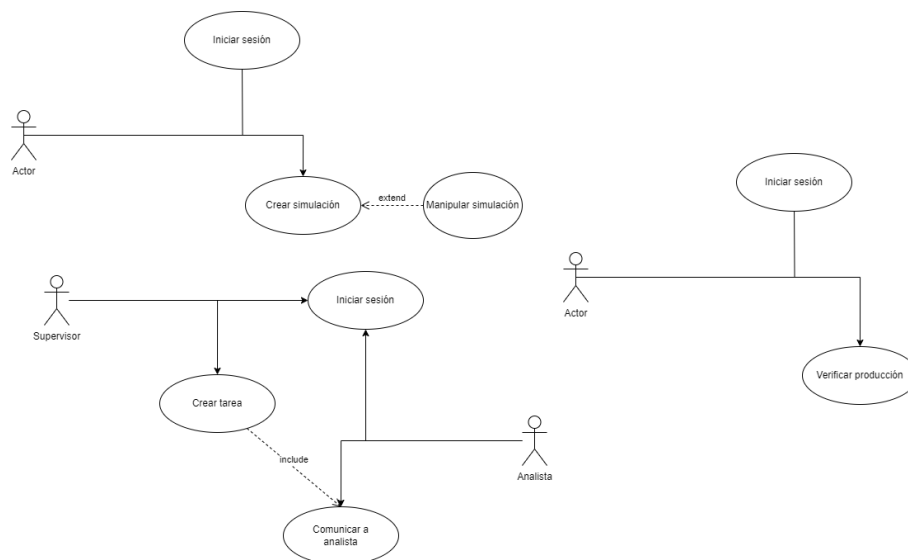


Figura 4.1: Diagrama de casos de uso"

CAPÍTULO 5

Solución Desarrollada

En este capítulo, presentaré una visión general de la solución desarrollada en el proyecto, tomando en cuenta la limitación de acceso a la base de datos (solo lectura de ciertas tablas). La solución tiene como objetivo principal mejorar la eficiencia en la producción de Ford Motors, permitiendo una monitorización en tiempo real y una optimización más efectiva del proceso de producción de vehículos. A continuación, describiré los componentes clave de la solución y cómo interactúan entre sí.

5.1 Visión General de la Solución

La solución desarrollada se centra en proporcionar herramientas de monitorización, simulación y gestión de producción para mejorar la eficiencia en la planta de Ford Motors. A pesar de las limitaciones de acceso a la base de datos, se lograron implementar varios componentes esenciales que permiten tomar decisiones más informadas y optimizar el proceso de producción. La visión general de la solución se ilustra en la Figura 5.1.

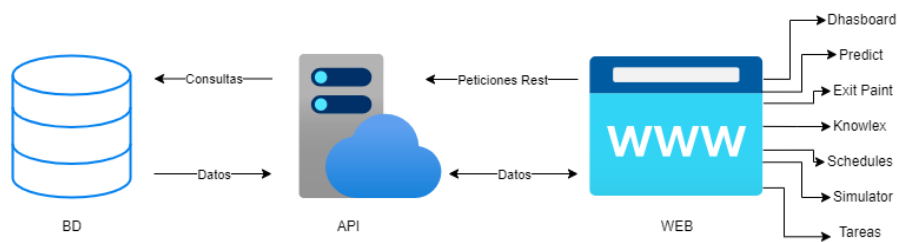


Figura 5.1: Arquitectura General de la Solución

La arquitectura general de la solución se compone de los siguientes elementos principales:

1. **Dashboard de Monitorización:** La interfaz de usuario principal de la solución es un dashboard de monitorización que proporciona una vista en tiempo real del estado de la fábrica de Ford Motors. Este dashboard muestra gráficos y tablas actualizados con información crítica, como la disponibilidad de patines y el progreso de la producción. Los supervisores y responsables pueden acceder a esta información para tomar decisiones informadas.
2. **Simulador de Producción:** La solución incluye un simulador de producción que permite a los usuarios recrear y evaluar diferentes escenarios y condiciones de pro-

ducción. Esto es crucial para la planificación estratégica y la optimización de recursos. Los usuarios pueden ajustar parámetros y analizar cómo afectarían los cambios al rendimiento de la fábrica.

3. **Predicción de Tiempo de Producción:** Se ha implementado un modelo de predicción impulsado por inteligencia artificial (IA) que puede estimar el tiempo requerido para que cada zona de la fábrica complete su proceso de producción. Esto permite una planificación más precisa y una asignación eficiente de recursos.
4. **Control Room de Ganancias/Pérdidas y Schedules con Gráficos:** La solución incorpora un control room en tiempo real que muestra un balance de ganancias o pérdidas en relación con los objetivos de producción establecidos. Esto permite a los responsables identificar desviaciones y tomar medidas correctivas oportunas. Además, se implementa una sección de schedules de producción que ofrece una visión actualizada de las actividades planificadas en la fábrica. Para un análisis más profundo de la producción, se generan gráficos basados en estos schedules. Ambas funcionalidades están integradas en la misma plataforma web para facilitar la gestión y la toma de decisiones en la producción.
5. **Predictor de Exit Paint:** Se ha desarrollado un predictor específico para la etapa de Exit Paint, que calcula los nuevos schedules basados en una fórmula distinta a los originales, utilizando datos del simulador. Esto proporciona una estimación más realista de la producción en esta etapa.
6. **Knowlex para la Detección de Paradas:** El Knowlex es un esquema de la planta que permite la detección de paradas en la producción. Aunque el acceso a la base de datos era limitado, se logró implementar esta funcionalidad para identificar paradas en la planta.
7. **Sistema de Gestión de Tareas y Comunicación Interna:** La solución incluye un sistema de gestión de tareas que permite a los coordinadores y supervisores asignar tareas específicas a los analistas. También se ha implementado un chat privado para facilitar la comunicación interna y la resolución rápida de problemas.

A pesar de las limitaciones de acceso a la base de datos, estos elementos de la solución proporcionan herramientas esenciales para la mejora de la producción y la toma de decisiones más informadas.

En la Figura 5.1, se muestra una representación visual de cómo se interconectan estos componentes en la arquitectura general de la solución. Aunque la comunicación con la base de datos es limitada, los componentes siguen interactuando entre sí para brindar funcionalidades esenciales.

5.2 Elementos de la Solución

La solución desarrollada se compone de varios elementos interconectados que trabajan en conjunto para mejorar la monitorización y optimización del proceso de producción en Ford Motors. A continuación, se detallan estos elementos:

5.2.1. Dashboard Principal

- **Descripción:** El Dashboard Principal es la puerta de entrada a la herramienta web. Proporciona una vista de alto nivel de la fábrica de coches de Ford, mostrando información esencial en tiempo real.

- **Visualización de Estado:** Presenta un esquema visual de la fábrica con indicadores de estado, como la disponibilidad de patines, la producción actual y las paradas detectadas.
- **Navegación Intuitiva:** Permite una navegación intuitiva a través de las diferentes secciones de la herramienta, incluyendo el simulador, el predictor, los schedules y el Control Room.

5.2.2. Simulador de Producción

- **Descripción:** El Simulador de Producción es una parte fundamental de la solución. Permite recrear el funcionamiento de la fábrica bajo diferentes condiciones y escenarios.
- **Escenarios Personalizables:** Ofrece la capacidad de configurar escenarios específicos, como cambios en la producción, para evaluar su impacto en tiempo real.
- **Análisis Predictivo:** Utiliza los schedules y datos en tiempo real para calcular cuantos coches habrá en cada zona a lo largo del día.

5.2.3. Predictor

- **Descripción:** Este módulo utiliza algoritmos avanzados de inteligencia artificial y aprendizaje automático para calcular y predecir el tiempo que tardara un determinado coche en pasar por cada zona de la factoría.
- **Entradas de Datos:** Utiliza una gran cantidad de datos previamente filtrados y preparados para entrenar los modelos con los que posteriormente se generaran las predicciones,
- **Planificación Precisa:** Facilita una planificación más precisa y una asignación adecuada de recursos en la fábrica ya que se puede determinar a que hora exacta estará un coche en una determinada zona de la factoría.

5.2.4. Exit Paint

- **Descripción:** Exit Paint es una herramienta diseñada para calcular nuevos schedules de producción basados en datos del simulador.
- **Fórmula Específica:** Utiliza una fórmula personalizada para estimar los nuevos schedules de producción, tomando en cuenta el volumen actual de modelos en la zona de Exit Paint.
- **Planificación Precisa:** Facilita una planificación más precisa y una asignación adecuada de recursos en la fábrica, generando schedules más realistas.

5.2.5. Knowlex (Esquema de Planta)

- **Descripción:** El Knowlex es un esquema de la planta de producción que desempeña un papel fundamental en la detección de paradas.
- **Detección de Paradas:** Utiliza los datos de los sensores para detectar paradas no programadas en la producción y genera alertas inmediatas.

- **Interfaz Intuitiva:** Proporciona una interfaz gráfica intuitiva para que los operadores y supervisores monitoreen el estado de la factoría sin tener que revisar diferentes webs.

5.2.6. Schedules y Control Room

- **Descripción:** La sección de Schedules y Control Room ofrece una visión integral de la producción planificada y el rendimiento en tiempo real.
- **Schedules de Producción:** Muestra los schedules actuales y futuros de producción en la fábrica, facilitando la coordinación y asignación de recursos.
- **Control Room de Ganancias/Pérdidas:** Proporciona indicadores clave de rendimiento (KPI) y alertas de desviación en tiempo real para evaluar el rendimiento en relación con los objetivos de producción y tomar medidas correctivas oportunas.

5.2.7. Gestión de Tareas y Comunicación Interna

- **Descripción:** La solución incluye un sistema de gestión de tareas que permite a los coordinadores y supervisores asignar tareas específicas a los analistas.
- **Chat Privado:** Facilita la comunicación interna a través de un chat privado, agilizando la resolución de problemas y la colaboración.
- **Seguimiento de Tareas:** Permite el seguimiento y la gestión eficiente de las tareas asignadas, garantizando una ejecución fluida de las actividades.

Estos elementos de la solución trabajan en conjunto para proporcionar una herramienta integral que mejora la monitorización, la toma de decisiones y la eficiencia en la producción de coches en Ford Motors.

CAPÍTULO 6

Diseño de la API REST

6.1 Introducción

Este capítulo detalla el diseño y desarrollo de la API REST (Interfaz de Programación de Aplicaciones Representacional), que es un componente crucial de la solución propuesta. La API REST facilita la comunicación entre el frontend de la aplicación web y la base de datos, permitiendo la interacción y el intercambio de datos entre los diversos módulos del sistema.

6.2 Tecnologías Utilizadas

El diseño de nuestra API involucra varias tecnologías esenciales cuidadosamente seleccionadas para garantizar un rendimiento óptimo y una gestión eficaz de los datos. Estas tecnologías incluyen:

- **Flask:** Flask es un micro marco de aplicaciones web de Python ampliamente reconocido y utilizado en la industria. Proporciona las herramientas necesarias para crear la estructura de la API, gestionar rutas y controladores, y facilitar la creación de puntos finales RESTful.
- **SQLAlchemy:** SQLAlchemy es una biblioteca de mapeo objeto-relacional (ORM) que simplifica la interacción con la base de datos. Permite definir modelos de datos en forma de clases Python, lo que facilita la manipulación de registros en la base de datos sin necesidad de escribir consultas SQL manualmente.
- **Flask-RESTful:** Flask-RESTful es una extensión de Flask que simplifica la creación de API RESTful. Proporciona una estructura organizada para definir recursos, gestionar rutas y manejar las solicitudes HTTP de manera eficiente.
- **PostgreSQL:** PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto. Se utiliza como el motor de base de datos principal para almacenar y recuperar datos de manera confiable y segura.

6.3 Estructura de la API

La API se organiza en varios componentes clave que garantizan una organización lógica y modular:

- **Rutas (Endpoints):** Las rutas definen las URL a las que los clientes pueden enviar solicitudes. Cada ruta se asocia con un recurso específico y se configura para manejar diferentes métodos HTTP, como GET, POST, PUT y DELETE. Por ejemplo, la ruta `/api/users` proporciona acceso a la información de usuarios.
- **Recursos:** Los recursos representan los objetos o datos manipulados por la API. Cada recurso tiene su propia URL única y está asociado con un conjunto de métodos HTTP que definen las operaciones permitidas en ese recurso. Ejemplos de recursos incluyen usuarios, tareas, horarios de producción, datos de simulación, etc.
- **Controladores:** Los controladores contienen la lógica para manejar las solicitudes entrantes a la API y administrar los modelos de datos asociados. Estos controladores se encargan de procesar las solicitudes, realizar validaciones, acceder a la base de datos y enviar las respuestas adecuadas al cliente.

6.4 Métodos HTTP

La API sigue las convenciones estándar de los métodos HTTP para realizar operaciones en los recursos:

- **GET:** Se utiliza para recuperar información sobre un recurso o una lista de recursos. Por ejemplo, se utiliza para obtener detalles de un usuario específico o una lista de tareas.
- **POST:** Se utiliza para crear un nuevo recurso en el servidor. Por ejemplo, se utiliza para agregar un nuevo usuario o una nueva tarea.
- **PUT:** Se utiliza para actualizar un recurso existente. Por ejemplo, se utiliza para modificar los datos de un usuario o una tarea existente.
- **DELETE:** Se utiliza para eliminar un recurso específico del servidor. Por ejemplo, se utiliza para eliminar un usuario o una tarea.

6.5 Respuestas JSON

Las respuestas generadas por la API se devuelven en formato JSON (JavaScript Object Notation). Este formato se utiliza ampliamente en el desarrollo web debido a su simplicidad y facilidad de lectura y procesamiento por parte de aplicaciones cliente y desarrolladores. Las respuestas JSON incluyen datos estructurados y, en muchos casos, información adicional sobre el estado de la solicitud.

6.6 Ejemplo de Endpoint y Respuesta

A continuación, se presentan ejemplos de endpoints de la API junto con ejemplos de las respuestas que generan:

- `/prediccion` [POST]: Este endpoint permite a los usuarios realizar predicciones basadas en ciertos datos de entrada. La respuesta incluye un JSON con la predicción y otra información relevante.

- / [GET]: Redirige a los usuarios a la página de inicio de sesión si aún no han iniciado sesión o a la página de inicio del panel de control si ya están autenticados.
- /sign_out [GET]: Cierra la sesión del usuario y lo redirige a la página de inicio de sesión.
- /profile/<user>[GET]: Muestra el perfil del usuario con el nombre de usuario especificado. La respuesta es una página HTML que contiene información detallada sobre el usuario.

```
1     # Importar la biblioteca os para manejar rutas de archivos
2 import os
3 # Importar la biblioteca json para trabajar con archivos JSON
4 import json
5
6 # Funcion para obtener el perfil de usuario
7 def profile(user):
8     # Obtener la ruta del archivo actual
9     current_dir = os.path.dirname(os.path.abspath(__file__))
10    # Construir la ruta relativa al archivo json
11    json_path = os.path.join(current_dir, 'static/data/users.json')
12
13    # Abrir el archivo JSON en modo lectura
14    with open(json_path, 'r') as f:
15        # Cargar los datos JSON en un diccionario
16        users = json.load(f)
17
18    # Buscar la informacion del usuario en el diccionario
19    user_info = users.get(user)
20
21    if user_info:
22        # Si se encuentra la informacion del usuario, renderizar un
23        # template HTML
24        return render_template('profile.html', username=user, user=
25        user_info, all_users=users)
26    else:
27        # Si el usuario no se encuentra, devolver un mensaje de error
28        return 'Usuario no encontrado'
```

Listing 6.1: Ejemplo de manejo de la solicitud GET en Flask

- /UltimaAct [GET]: Devuelve la fecha y hora de la última actualización de los datos almacenados en la base de datos.
- /graficas/<username>[POST]: Permite a los usuarios enviar datos de gráficos desde el cliente. No devuelve una respuesta explícita, ya que su función principal es actualizar los datos de gráficos en el servidor.

Diseño y Desarrollo del Dashboard

7.1 Tecnologías y Bibliotecas Utilizadas

El desarrollo del dashboard involucró la implementación de diversas tecnologías y bibliotecas clave que contribuyeron a su funcionalidad y apariencia visual.

7.1.1. Bootstrap Icons

Bootstrap Icons es una biblioteca que proporciona una extensa colección de iconos de alta calidad específicamente diseñados para su uso con el framework Bootstrap. Estos iconos desempeñaron un papel esencial para mejorar la interfaz de usuario, facilitando la identificación de funciones y características.

7.1.2. jQuery Slim

jQuery es una conocida biblioteca de JavaScript que simplifica la manipulación del DOM (Document Object Model) y la interacción con el usuario en páginas web. Se optó por la versión "Slim" de jQuery, una variante ligera que incluye solo las características esenciales. jQuery se utilizó para agregar interactividad a la aplicación, lo que permitió a los usuarios interactuar con la interfaz de manera dinámica y recibir actualizaciones en tiempo real.

7.1.3. Popper.js

Popper.js es una biblioteca fundamental para posicionar elementos de manera dinámica y precisa en la interfaz de usuario. Trabajando en conjunto con Bootstrap, mejoró significativamente el diseño y la disposición de los elementos en la página, asegurando una presentación óptima en diferentes tamaños de pantalla y dispositivos.

7.1.4. Bootstrap

Bootstrap es un framework de código abierto ampliamente utilizado que simplifica la creación de diseños y componentes de interfaz web consistentes y receptivos. En este proyecto, Bootstrap desempeñó un papel fundamental al agilizar el proceso de diseño y garantizar una apariencia coherente en diversos dispositivos, desde computadoras de escritorio hasta dispositivos móviles.

7.1.5. Dashboard Stylesheet

Se diseñó una hoja de estilo personalizada (CSS) específicamente para el panel de control de la aplicación. Esta hoja de estilo definió la apariencia de los elementos en la interfaz, asegurando una presentación visualmente atractiva y coherente con la identidad de la marca. Incluyó la selección de colores, fuentes, espaciado y disposición de elementos.

7.1.6. GoJS

GoJS es una biblioteca especializada en la creación de diagramas interactivos y gráficos de manera programática. En este proyecto, GoJS se utilizó para representar visualmente datos complejos y relaciones en forma de diagramas interactivos. Esto permitió a los usuarios comprender mejor la información a través de representaciones visuales claras y dinámicas.

7.1.7. Moment.js

Moment.js es una biblioteca de JavaScript que simplifica el manejo de fechas y horas. Facilita tareas como el formateo, la manipulación y los cálculos relacionados con el tiempo. En el contexto de la aplicación, Moment.js se utilizó para trabajar con fechas y horas relacionadas con la producción y los horarios.

7.1.8. Highcharts

Highcharts es una biblioteca de gráficos que permite la creación de visualizaciones interactivas y atractivas de datos en páginas web. En este proyecto, Highcharts se empleó para representar datos numéricos y estadísticas en forma de gráficos, facilitando la visualización de tendencias y patrones clave.

7.1.9. Highcharts Exporting y Highcharts Export Data

Estos módulos adicionales de **Highcharts** agregan funcionalidades de exportación de gráficos y datos a diferentes formatos, como PNG o PDF. Esto permite a los usuarios guardar o compartir visualizaciones de datos y datos subyacentes de gráficos de manera conveniente y eficiente.

7.1.10. Highcharts Accessibility

Highcharts Accessibility es un módulo especializado que mejora la accesibilidad de los gráficos. Garantiza que los gráficos sean comprensibles para usuarios con discapacidades visuales o cognitivas, cumpliendo con estándares de accesibilidad web y garantizando que la información sea accesible para todos los usuarios.

7.2 Diseño del Dashboard

El dashboard se diseñó con un enfoque en la eficacia y la facilidad de uso. Se dividió en secciones lógicas que presentan datos y herramientas de manera clara y accesible para los usuarios. Cada sección se benefició de las bibliotecas y tecnologías mencionadas anteriormente para garantizar una experiencia fluida y atractiva.

7.3 Componentes Destacados del Dashboard

El dashboard incluye varios componentes destacados que permiten a los usuarios interactuar con los datos y tomar decisiones informadas. Algunos de estos componentes incluyen:

7.3.1. Gráficos Interactivos

Se utilizaron gráficos interactivos de **Highcharts** para representar visualmente datos numéricos y tendencias clave. Estos gráficos permiten a los usuarios explorar datos en detalle, ajustar rangos de fechas y obtener información instantánea al pasar el cursor sobre puntos de datos.

7.3.2. Diagramas Dinámicos

Los diagramas dinámicos generados con **GoJS** ofrecen una representación visual de datos complejos y relaciones entre elementos. Los usuarios pueden explorar y manipular estos diagramas para comprender mejor la información y tomar decisiones basadas en datos.

7.3.3. Filtros y Controles

Se incorporaron filtros y controles interactivos para permitir a los usuarios personalizar su experiencia. Estos elementos facilitan la selección de fechas, la elección de modelos o la visualización de datos específicos, lo que garantiza que los usuarios puedan acceder a la información que necesitan de manera eficiente.

7.3.4. Exportación de Datos

La funcionalidad de exportación de datos permite a los usuarios guardar gráficos y datos en formatos como PNG o PDF incluso Excel. Esto facilita la generación de informes personalizados y la compartición de datos relevantes con otros miembros del equipo.

7.4 Bocetos de las Interfaces de la Aplicación

Antes de comenzar con el desarrollo completo de la aplicación web, se crearon bocetos a mano para definir las interfaces de usuario y la estructura general de la herramienta. Estos bocetos se compartieron con el supervisor y el coordinador para obtener su aprobación y realizar ajustes según sus comentarios.

A continuación, se presentan algunos bocetos clave de las interfaces de la aplicación:

1. **Dashboard:** El dashboard principal muestra una representación visual de la fábrica con información en tiempo real sobre la producción en cada zona, recursos disponibles y posibles paradas. Los gráficos y tablas proporcionan una visión general de la eficiencia de la producción (ver Figura 7.1).
2. **Simulador:** La interfaz del simulador permite a los usuarios ajustar la cantidad de coches producidos en diferentes zonas y evaluar el rendimiento resultante. Los resultados de las simulaciones se presentan en gráficos y tablas para una fácil interpretación (ver Figura 7.2).

3. **Predictor:** La interfaz del predictor solicita el VIN de un coche y muestra el tiempo estimado de producción en cada zona de la fábrica. Los datos se presentan de manera clara y detallada (ver Figura 7.3).
4. **Schedules:** La página de schedules muestra los programas de producción actuales y permite a los usuarios realizar ajustes y actualizaciones según sea necesario (ver Figura 7.4).
5. **Control Room:** El control room presenta un resumen en tiempo real del progreso de la producción en comparación con los schedules planificados. Cualquier desviación significativa se resalta para una rápida identificación y toma de decisiones (ver Figura 7.5).
6. **Tareas:** La interfaz de gestión de tareas permite al coordinador asignar tareas específicas a los analistas y supervisores. Además, incluye un chat privado para facilitar la comunicación interna (ver Figura 7.6).

Estos bocetos fueron la base para el diseño final de las interfaces de usuario de la aplicación web, asegurando que cumpliera con las necesidades y expectativas de los usuarios y proporcionando una experiencia de usuario intuitiva y eficiente.

En el próximo capítulo, se detallará el proceso de desarrollo de la solución, incluyendo la arquitectura de la aplicación, el contexto tecnológico utilizado y ejemplos de código para ilustrar la implementación de algunas funcionalidades clave.

7.5 Accesibilidad y Responsividad

El diseño del dashboard se realizó con un enfoque en la accesibilidad, asegurando que los gráficos y la información sean comprensibles para todos los usuarios, incluidos aquellos con discapacidades visuales o cognitivas. Además, el dashboard es altamente receptivo, lo que significa que se adapta de manera efectiva a diferentes tamaños de pantalla y dispositivos, garantizando una experiencia de usuario consistente.

7.6 Conclusión

El dashboard desarrollado para esta aplicación es un componente central que permite a los usuarios interactuar de manera efectiva con los datos y tomar decisiones informadas. Gracias a las tecnologías y bibliotecas mencionadas, se logró una interfaz visualmente atractiva y altamente funcional que mejora la productividad y la toma de decisiones dentro de la organización.

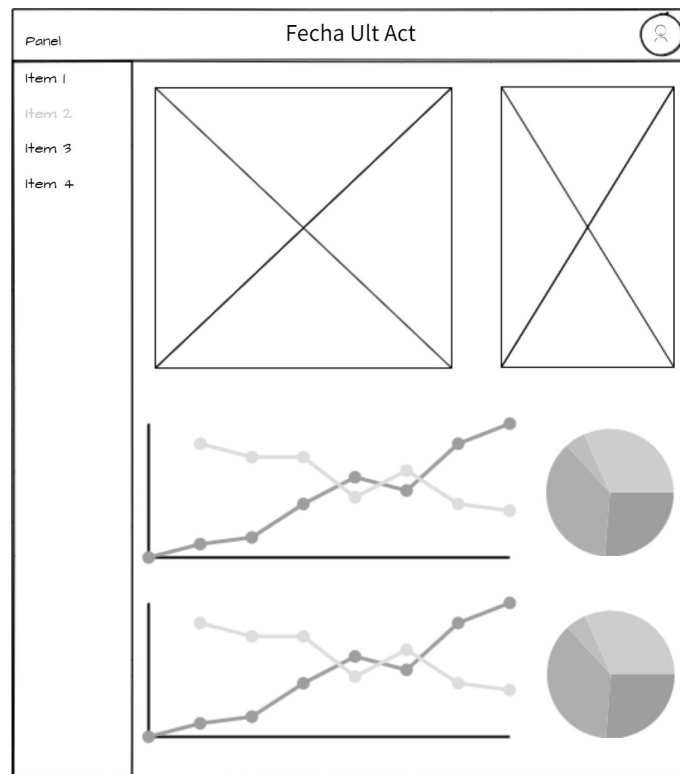


Figura 7.1: Boceto del Dashboard

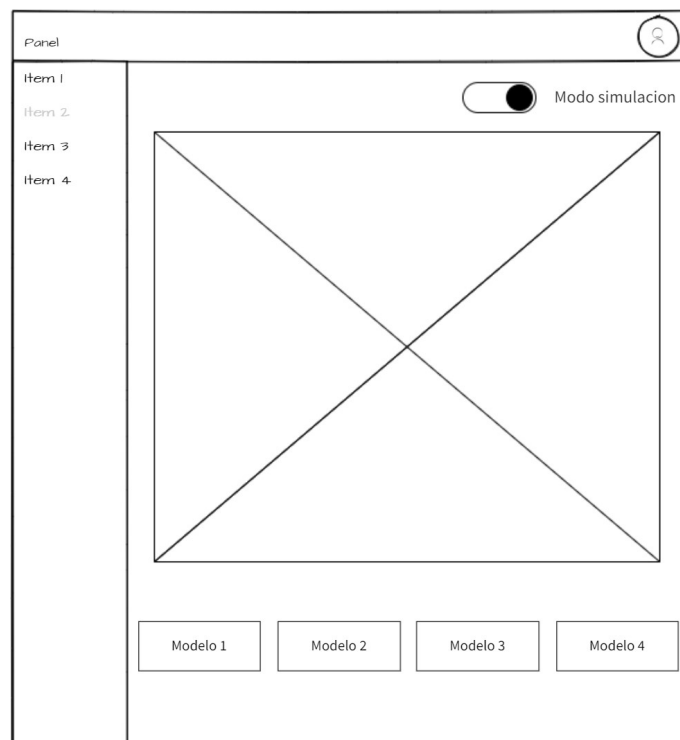


Figura 7.2: Boceto del Simulador

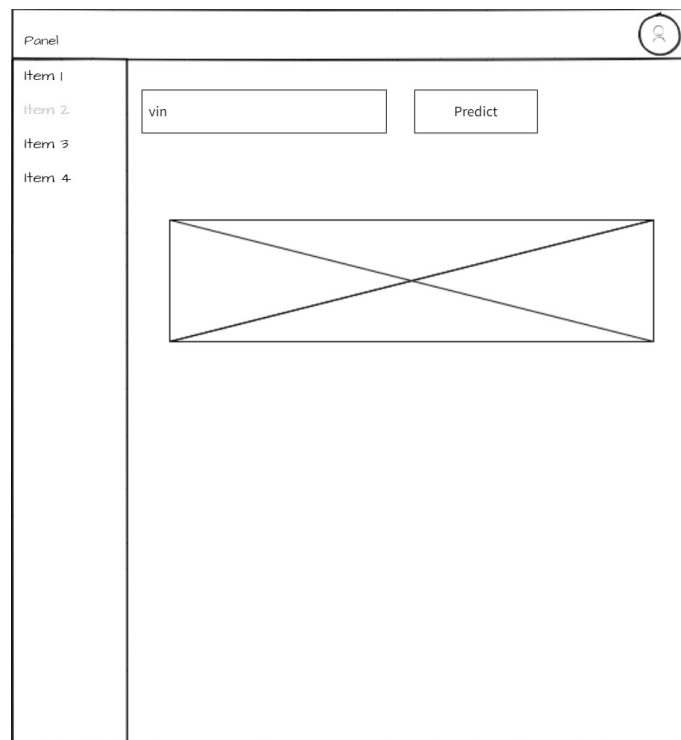


Figura 7.3: Boceto del Predictor

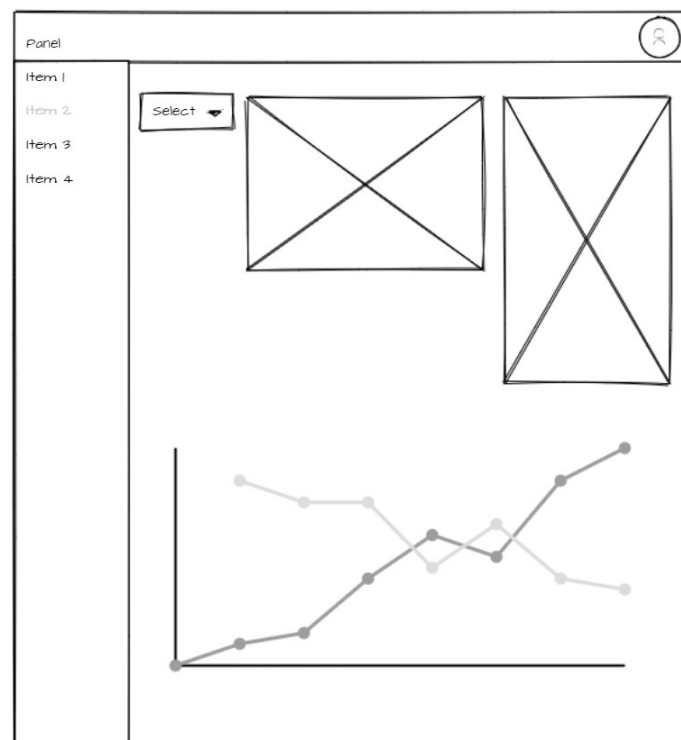


Figura 7.4: Boceto de Schedules

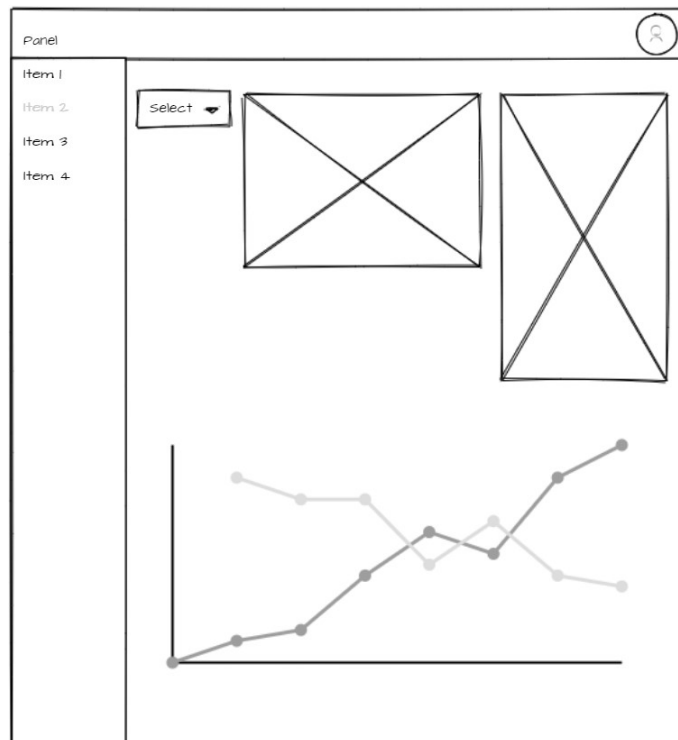


Figura 7.5: Boceto de Control Room

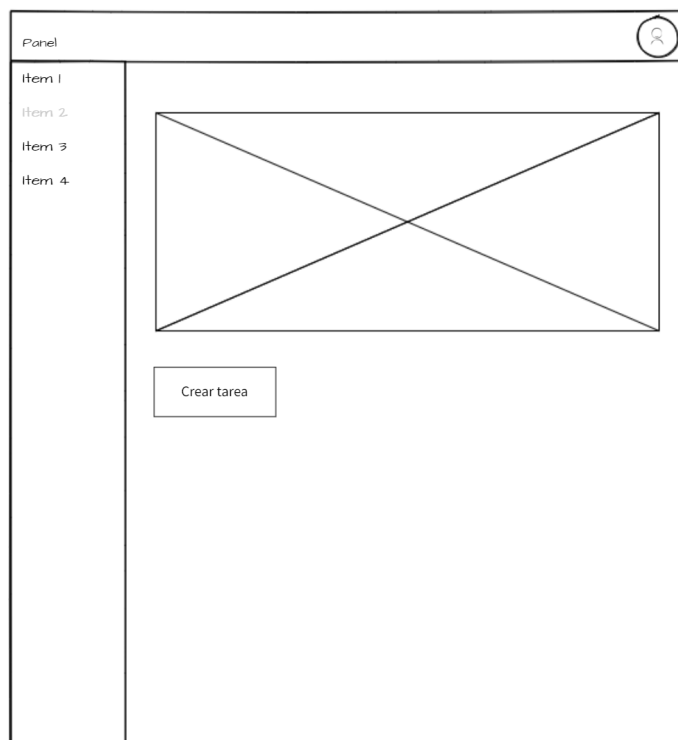


Figura 7.6: Boceto de Gestión de Tareas

CAPÍTULO 8

Desarrollo de la Solución

En este capítulo se describe en detalle el proceso de desarrollo de la aplicación web de seguimiento y optimización de producción en la fábrica. Se abordarán aspectos clave como la arquitectura general, el contexto tecnológico y se proporcionarán ejemplos de código representativos para ilustrar la implementación de funcionalidades esenciales.

8.1 Arquitectura del Sistema

En este capítulo, exploraré en profundidad la arquitectura subyacente de la aplicación web de gestión de tareas. Describiremos las tecnologías utilizadas, los componentes clave y la interacción entre ellos. La arquitectura se ha diseñado cuidadosamente para proporcionar una experiencia fluida y segura a los usuarios en la gestión de sus tareas y la comunicación en tiempo real. A medida que profundizamos en la arquitectura, veremos cómo los diferentes elementos se entrelazan para crear una plataforma completa y funcional.

8.1.1 Componentes de la Arquitectura

La arquitectura de la aplicación se compone de varios elementos interconectados que trabajan en conjunto para brindar una experiencia completa y coherente a los usuarios. A continuación, se describen los componentes clave de la arquitectura:

Servidor y Backend

El servidor está construido utilizando Flask, que actúa como el núcleo del backend. Maneja las solicitudes entrantes de los usuarios, procesa la lógica comercial y responde con datos y vistas renderizadas. El backend se encarga de autenticar y autorizar a los usuarios, gestionar tareas y mensajes, y coordinar la interacción entre el frontend y los archivos JSON donde se almacenan los datos.

Frontend y Plantillas

El frontend se construye principalmente utilizando HTML, CSS y JavaScript. Flask utiliza plantillas para renderizar dinámicamente las páginas web. Los usuarios interactúan con la interfaz de usuario, marcan subtareas como completadas, envían mensajes en el chat y visualizan las tareas asignadas y creadas. El frontend también maneja la presentación de datos y la generación de miniaturas para los archivos adjuntos.

Almacenamiento de Datos

Los datos de usuarios, tareas y mensajes se almacenan en archivos JSON. Estos archivos JSON se utilizan como una forma flexible y accesible de almacenar información. El backend carga, actualiza y guarda datos en estos archivos, manteniendo la coherencia y persistencia de la información en toda la aplicación.

8.1.2. Comunicación en Tiempo Real

La aplicación permite la comunicación en tiempo real entre usuarios a través del sistema de chat. Cuando los usuarios envían mensajes o adjuntan archivos en el chat de una tarea, la información se actualiza y muestra instantáneamente a los demás usuarios involucrados en esa tarea.

8.1.3. Interacción y Flujo del Usuario

El flujo del usuario en la aplicación comienza con la autenticación, seguida de la visualización de la página de tareas personalizada. Los usuarios pueden ver sus tareas asignadas y creadas, marcar subtareas como completadas, comunicarse a través del chat y visualizar detalles de las tareas.

1. **Inicio de Sesión y Autenticación:** Los usuarios inician sesión utilizando sus credenciales. Flask maneja la autenticación y autorización para garantizar el acceso seguro a las páginas de tareas.
2. **Página de Tareas Personalizada:** Una vez autenticados, los usuarios son redirigidos a su página de tareas. Aquí, pueden ver dos secciones: "Tareas Asignadas" y "Tareas Creadas".
3. **Gestión de Tareas:** Los usuarios pueden ver detalles de tareas, marcar subtareas como completadas y enviar mensajes en el chat asociado a cada tarea.
4. **Comunicación en Tiempo Real:** Los mensajes enviados en el chat se actualizan instantáneamente para todos los usuarios involucrados en la tarea, permitiendo una comunicación fluida.

8.1.4. Seguridad y Privacidad

La seguridad y la privacidad de los datos y la comunicación son fundamentales en la arquitectura de la aplicación. Se manejan aspectos como la encriptación de contraseñas, la autenticación de usuarios y la protección de datos sensibles. Además, los archivos adjuntos se gestionan de manera segura y se generan miniaturas para su visualización.

8.1.5. Escalabilidad y Mejoras Futuras

La arquitectura está diseñada para ser modular y escalable, lo que permite realizar mejoras y agregar nuevas características en el futuro. Por ejemplo, se podrían implementar notificaciones en tiempo real para informar a los usuarios sobre cambios en las tareas, o se podría optimizar el rendimiento para manejar una mayor cantidad de usuarios y tareas.

8.1.6. Conclusiones

La arquitectura de la aplicación de gestión de tareas combina tecnologías modernas, componentes interconectados y un flujo de usuario intuitivo para brindar a los usuarios una plataforma efectiva y eficiente para gestionar sus tareas y comunicarse en tiempo real. La seguridad, la usabilidad y la escalabilidad han sido consideradas en cada etapa del desarrollo para garantizar una experiencia de usuario excepcional.

8.2 Ejemplos de Código

A continuación, se presentan ejemplos de código que ilustran aspectos clave de la implementación de la aplicación:

8.2.1. Ejemplos de Endpoints y Funciones

En esta sección, proporcionaré ejemplos de código para varios endpoints y funciones utilizados en nuestro sistema.

Obtener Tareas Asignadas (GET /tareasUser/<username>)

El siguiente ejemplo muestra cómo obtener todas las tareas que han sido asignadas al usuario especificado utilizando el endpoint /tareasUser/<username>.

```

1 @app.route('/tareasUser/<username>', methods=['GET'])
2 def tareasUser(username):
3     current_dir = os.path.dirname(os.path.abspath(__file__))
4     json_path = os.path.join(current_dir, 'static/data/tareas.json')
5
6     with open(json_path, 'r') as f:
7         tasks = json.load(f)
8
9     assigned_tasks = []
10    for user_tasks in tasks.values():
11        for task in user_tasks:
12            if task['assigned_to'] == username or task['created_by'] ==
13                username:
14                assigned_tasks.append(task)
15
16    return jsonify(assigned_tasks)

```

Listing 8.1: Endpoint para Obtener Tareas Asignadas

En este código, defino un endpoint /tareasUser/<username> que toma el nombre de usuario como parámetro. Luego, leo las tareas desde el archivo JSON y filtro las tareas asignadas al usuario especificado.

Enviar Mensaje en el Chat de una Tarea (POST /tareas/<username>/<task_id>/chat)

Este ejemplo muestra cómo enviar un mensaje en el chat de una tarea utilizando el endpoint /tareas/<username>/<task_id>/chat.

```

1 @app.route('/tareas/<username>/<task_id>/chat', methods=['POST'])
2 def send_message(username, task_id):
3     # Obtener la información del mensaje desde el formulario
4     sender = request.form['sender']
5     content = request.form['content']

```

```

6 timestamp = datetime.now().isoformat()
7
8 # Generar un ID de mensaje nico utilizando uuid.uuid4()
9 message_id = str(uuid.uuid4())
10
11 # Construir el objeto de mensaje
12 message = {
13     'id': message_id,
14     'sender': sender,
15     'content': content,
16     'timestamp': timestamp,
17     'attachments': [],
18     'read': False
19 }
20
21 current_dir = os.path.dirname(os.path.abspath(__file__))
22 for attachment in request.files.getlist('attachment'):
23     filename = secure_filename(attachment.filename)
24     attachment.save(os.path.join('static/data/Chats', task_id, filename))
25     attachment_url = url_for('static', filename=f'data/Chats/{task_id}/{
26         filename}')
27     message['attachments'].append(f'{filename}|{attachment_url}')
28
29 json_path = os.path.join(current_dir, 'static/data/tareas.json')
30
31 # Cargar las tareas existentes del archivo JSON
32 with open(json_path, 'r') as f:
33     tasks = json.load(f)
34
35 # Buscar la tarea correspondiente y agregar el mensaje al chat
36 for user, user_tasks in tasks.items():
37     for task in user_tasks:
38         if task.get('id') == task_id:
39             task['messages'].append(message)
40             break
41
42 # Guardar las tareas actualizadas en el archivo JSON
43 with open(json_path, 'w') as f:
44     json.dump(tasks, f)
45
46 return redirect(url_for('task_details', username=username, id=task_id))

```

Listing 8.2: Endpoint para Enviar Mensaje en el Chat de una Tarea

En este código, defino un endpoint `/tareas/<username>/<task_id>/chat` para enviar mensajes en el chat de una tarea. El mensaje incluye información sobre el remitente, el contenido y las posibles adjuntos.

Eliminar una Tarea (DELETE `/tasks/<task_id>`)

El siguiente ejemplo muestra cómo eliminar una tarea utilizando el endpoint `/tasks/<task_id>`.

```

1 @app.route('/tasks/<task_id>', methods=['DELETE'])
2 def delete_task(task_id):
3     # Cargar las tareas desde el archivo JSON
4     current_dir = os.path.dirname(os.path.abspath(__file__))
5     json_path = os.path.join(current_dir, 'static/data/tareas.json')
6
7     with open(json_path, 'r') as f:
8         data = json.load(f)
9
10    # Encontrar el usuario que creó la tarea

```

```

11 user = next((user for user in data if any(task['id'] == task_id for task in
12     data[user])), None)
13 # Encontrar y eliminar la tarea de la lista de tareas del usuario
14 if user:
15     tasks = data[user]
16     task = next((task for task in tasks if task['id'] == task_id), None)
17     if task:
18         tasks.remove(task)
19
20     # Eliminar al usuario de los datos si ya no tiene m s tareas
21     if not tasks:
22         del data[user]
23
24     # Guardar los datos actualizados en el archivo JSON
25     with open(json_path, 'w') as f:
26         if not data:
27             json.dump({}, f)
28         else:
29             json.dump(data, f)
30
31     # Eliminar la carpeta asociada del sistema de archivos
32     folder_path = os.path.join('static/data/Chats', str(task_id))
33     shutil.rmtree(folder_path)
34
35 # Devolver una respuesta de xito
36 return '', 204

```

Listing 8.3: Endpoint para Eliminar una Tarea

Este código define un endpoint `/tasks/<task_id>` para eliminar una tarea. El proceso incluye la eliminación de la tarea de la lista de tareas del usuario y la eliminación de la carpeta asociada en el sistema de archivos.

8.3 Ejemplos de la pagina de Tareas

En esta sección, se presentan ejemplos representativos de la pagina de tareas (*Tareas*). Estos ejemplos incluyen fragmentos de código HTML, CSS y JavaScript utilizados para implementar funcionalidades específicas en el panel de control de tareas.

8.3.1. Código HTML, CSS y JavaScript del Panel de Control de Tareas

A continuación, mostraré un extracto de código HTML, CSS y JavaScript que se utiliza para crear y mostrar las tareas en el panel de control. Este código incluye elementos de diseño, tablas y formularios interactivos.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <!-- Encabezado y enlaces a bibliotecas externas -->
5     <!-- ... -->
6 </head>
7 <body>
8     <!-- Contenedor principal -->
9     <div class="container-fluid bg-white">
10         <!-- Encabezado de navegaci n -->
11         <nav class="navbar navbar-dark fixed-top bg-primary flex-md-nowrap p
12             -0">
13             <!-- Logo y enlaces de navegaci n -->

```

```

13     <!-- ... -->
14 </nav>
15
16 <!-- Contenido del panel de control -->
17 <div class="row">
18     <!-- Barra lateral de navegaci n -->
19     <nav class="col-sm-3 col-md-2 bg-light sidebar">
20         <!-- Elementos de navegaci n -->
21         <!-- ... -->
22     </nav>
23
24     <!-- Contenido principal -->
25     <div class="col-sm-9 col-md-10 bg-white mt-5">
26         <div class="container">
27             <!-- Tablas de tareas asignadas y enviadas -->
28             <ul class="nav nav-tabs">
29                 <!-- ... -->
30             </ul>
31             <div class="tab-content">
32                 <div class="tab-pane container active" id="Tareas
33                     enviadas">
34                     <!-- Tabla de tareas enviadas -->
35                     <table class="table">
36                         <!-- Encabezados de columna -->
37                         <!-- Filas de datos -->
38                     </table>
39                 <div class="tab-pane container" id="Tareas asignadas
40                     ">
41                     <!-- Tabla de tareas asignadas -->
42                     <table class="table">
43                         <!-- Encabezados de columna -->
44                         <!-- Filas de datos -->
45                     </table>
46                 </div>
47             </div>
48
49             <!-- Formulario para agregar tarea (si es aplicable) -->
50             <button id="showFormButton" class="btn btn-primary"
51                 onclick="mostrarForm()">Agregar Tarea</button>
52             <div id="taskFormCard" class="card">
53                 <div class="card-body">
54                     <!-- Formulario para agregar tarea -->
55                     <form action="/tareas/{{ username }}" method="
56                         POST">
57                         <!-- Campos del formulario -->
58                     </form>
59                 </div>
60             </div>
61         </div>
62
63     <!-- Scripts JavaScript utilizados -->
64     <script>
65         // Funciones y l gica JavaScript
66         // ...
67     </script>
68 </body>
69 </html>

```

Listing 8.4: Fragmento de código HTML para el Panel de Control de Tareas

8.3.2. Ejemplos Representativos de Otras Partes del Sistema

Además del panel de control, el sistema incluye otras partes, como interfaces de usuario y conexión a la API. A continuación, proporcionaré ejemplos representativos de estas partes del sistema.

Interfaz de Usuario

Aquí presento un fragmento de código HTML y CSS que representa una interfaz de usuario para la visualización de tareas asignadas y enviadas. Este ejemplo incluye elementos de diseño y estilos para una experiencia de usuario agradable.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <!-- Encabezado y enlaces a bibliotecas externas -->
5   <!-- ... -->
6 </head>
7 <body>
8   <!-- Contenedor principal de la interfaz de usuario -->
9   <div class="user-interface-container">
10    <!-- Encabezado de la interfaz de usuario -->
11    <header class="user-interface-header">
12      <!-- Logo y titulo -->
13      <!-- ... -->
14    </header>
15
16    <!-- Contenido principal de la interfaz de usuario -->
17    <main class="user-interface-main">
18      <!-- Visualización de tareas asignadas y enviadas -->
19      <div class="task-display">
20        <!-- Tareas asignadas -->
21        <div class="assigned-tasks">
22          <!-- Lista de tareas asignadas -->
23          <!-- ... -->
24        </div>
25
26        <!-- Tareas enviadas -->
27        <div class="sent-tasks">
28          <!-- Lista de tareas enviadas -->
29          <!-- ... -->
30        </div>
31      </div>
32    </main>
33
34    <!-- Pie de página de la interfaz de usuario -->
35    <footer class="user-interface-footer">
36      <!-- Información de contacto y enlaces -->
37      <!-- ... -->
38    </footer>
39  </div>
40
41  <!-- Estilos CSS para la interfaz de usuario -->
42  <style>
43    /* Estilos para el contenedor principal */
44    .user-interface-container {
45      /* ... */
46    }
47
48    /* Estilos para el encabezado */
49    .user-interface-header {
50      /* ... */

```

```

51     }
52
53     /* Estilos para el contenido principal */
54     .user-interface-main {
55         /* ... */
56     }
57
58     /* Estilos para la visualización de tareas */
59     .task-display {
60         /* ... */
61     }
62
63     /* Estilos para las listas de tareas */
64     .assigned-tasks, .sent-tasks {
65         /* ... */
66     }
67
68     /* Estilos adicionales */
69     /* ... */
70 </style>
71 </body>
72 </html>

```

Listing 8.5: Interfaz de Usuario para Visualización de Tareas

Conexión a la API

A continuación, presento un ejemplo de cómo se puede realizar una conexión a la API de nuestro sistema utilizando JavaScript. En este caso, se muestra cómo obtener la lista de tareas asignadas desde el servidor y mostrarla en la interfaz de usuario.

```

1 // Realizar una solicitud GET a la API para obtener tareas asignadas
2 fetch('/api/tasks/assigned')
3   .then(response => response.json())
4   .then(data => {
5       // Obtener el contenedor de tareas asignadas
6       const assignedTasksContainer = document.querySelector('.assigned-tasks')
7
8       // Generar elementos de lista para cada tarea y agregarlos al
9       // contenedor
10      data.forEach(task => {
11          const taskItem = document.createElement('li');
12          taskItem.textContent = task.title;
13          assignedTasksContainer.appendChild(taskItem);
14      });
15      .catch(error => {
16          console.error('Error al obtener tareas asignadas:', error);
17      });

```

Listing 8.6: Conexión a la API para Obtener Tareas Asignadas

En esta sección, hemos presentado ejemplos de código representativos para el panel de control de tareas y otras partes del sistema, como la interfaz de usuario y la conexión a la API. Estos ejemplos ilustran cómo se pueden implementar y utilizar diferentes componentes y funcionalidades en nuestro sistema.

8.3.3. Estructura del JSON de Usuarios

El archivo JSON de usuarios contiene la información de los usuarios que tienen acceso al sistema. A continuación se muestra un ejemplo modificado con nombres y contraseñas aleatorias por cuestiones de seguridad:

```
{
  'user123': {
    'password': 'randompassword123',
    'role': 'admin'
  },
  'john.doe': {
    'password': 'securepass456',
    'role': 'supervisor'
  },
  'jane.smith': {
    'password': 'P@ssw0rd789',
    'role': 'supervisor'
  },
  'alice.jones': {
    'password': 's3cur3password',
    'role': 'coordinator'
  },
  'bob.miller': {
    'password': 'strongpass987',
    'role': 'coordinator'
  },
  ... (otros usuarios)
  'emma.robinson': {
    'password': 'safepass123',
    'role': 'analyst'
  }
}
```

Cada entrada en el JSON representa un usuario y contiene los siguientes campos:

- **username:** Nombre de usuario único como clave.
- **password:** Contraseña del usuario (las contraseñas han sido modificadas por cuestiones de seguridad).
- **role:** Rol del usuario en el sistema (admin, supervisor, coordinator o analyst).

Esta estructura permite gestionar los roles y permisos de los usuarios en el sistema de manera eficiente. Las contraseñas en este ejemplo han sido modificadas y no reflejan contraseñas reales utilizadas en un sistema en producción.

8.3.4. Ejemplo de JSON de Tareas

El archivo JSON de tareas contiene la información sobre las tareas asignadas a diferentes usuarios en el sistema. A continuación se muestra un ejemplo modificado con datos aleatorios por cuestiones de seguridad:

```

{
  'user123': [{
    'id': 'a1bded68-fb08-4d56-ae57-56908ca81cd5',
    'title': 'Tarea de ejemplo 1',
    'description': 'Esta es una tarea de ejemplo',
    'created_by': 'user123',
    'created_at': '2023-06-29',
    'due_date': '',
    'priority': 'Alta',
    'assigned_to': 'user123',
    'subtasks': [
      {'title': 'Subtarea 1', 'completed': true},
      {'title': 'Subtarea 2', 'completed': false},
      {'title': 'Subtarea 3', 'completed': false}
    ],
    'messages': [
      {'id': 'b420dd9b-dbd5-437b-a18f-6eb000936ad4', 'sender': 'user123',
        'content': '¡Hola!', 'timestamp': '2023-06-29T11:55:58.273881',
        'attachments': [], 'read': false}
    ]
  }],
  'john.doe': [{
    'id': 'ccae1a68-1057-4408-847c-e9de5e51d9b8',
    'title': 'Tarea importante',
    'description': 'Esta tarea es muy importante',
    'created_by': 'john.doe',
    'created_at': '2023-06-30',
    'due_date': '2023-07-05',
    'priority': 'Alta',
    'assigned_to': 'jane.smith',
    'subtasks': [
      {'title': 'Subtarea 1', 'completed': false},
      {'title': 'Subtarea 2', 'completed': true}
    ],
    'messages': [
      {'id': 'a00fd282-986b-417c-84b8-a0a9fbc43784', 'sender': 'john.doe',
        'content': '¡Hola!', 'timestamp': '2023-06-30T13:36:55.583715',
        'attachments': [], 'read': false},
      {'id': 'c8c013cd-6cc4-450c-ab11-16fe4bb16187', 'sender': 'john.doe',
        'content': '¡Cómo estás?', 'timestamp': '2023-07-01T08:30:32.481013',
        'attachments': [], 'read': false}
    ]
  }],
  ... (otras tareas)
}

```

Cada entrada en el JSON representa una tarea y contiene los siguientes campos:

- id: ID único de la tarea.
- title: Título de la tarea.
- description: Descripción detallada de la tarea.

- `created_by`: Nombre de usuario que creó la tarea.
- `created_at`: Fecha de creación de la tarea.
- `due_date`: Fecha límite para completar la tarea (opcional).
- `priority`: Prioridad de la tarea.
- `assigned_to`: Nombre de usuario al que se asignó la tarea.
- `subtasks`: Lista de subtareas asociadas a la tarea.
- `messages`: Lista de mensajes relacionados con la tarea.

Esta estructura permite almacenar y gestionar la información de las tareas y su progreso en el sistema. Los datos en este ejemplo han sido modificados y no reflejan tareas reales en un sistema en producción.

CAPÍTULO 9

Producto desarrollado

9.1 Presentación de la Aplicación

La aplicación que se describe en este capítulo es una herramienta web diseñada para la gestión y supervisión de la producción en un entorno industrial. Permite a los usuarios llevar a cabo diversas tareas clave para mantener un flujo de producción eficiente y resolver problemas en tiempo real.

9.1.1. Interfaz de Inicio de Sesión

Para comenzar a utilizar la aplicación, el usuario debe iniciar sesión con sus credenciales. A continuación, se muestra la pantalla de inicio de sesión:

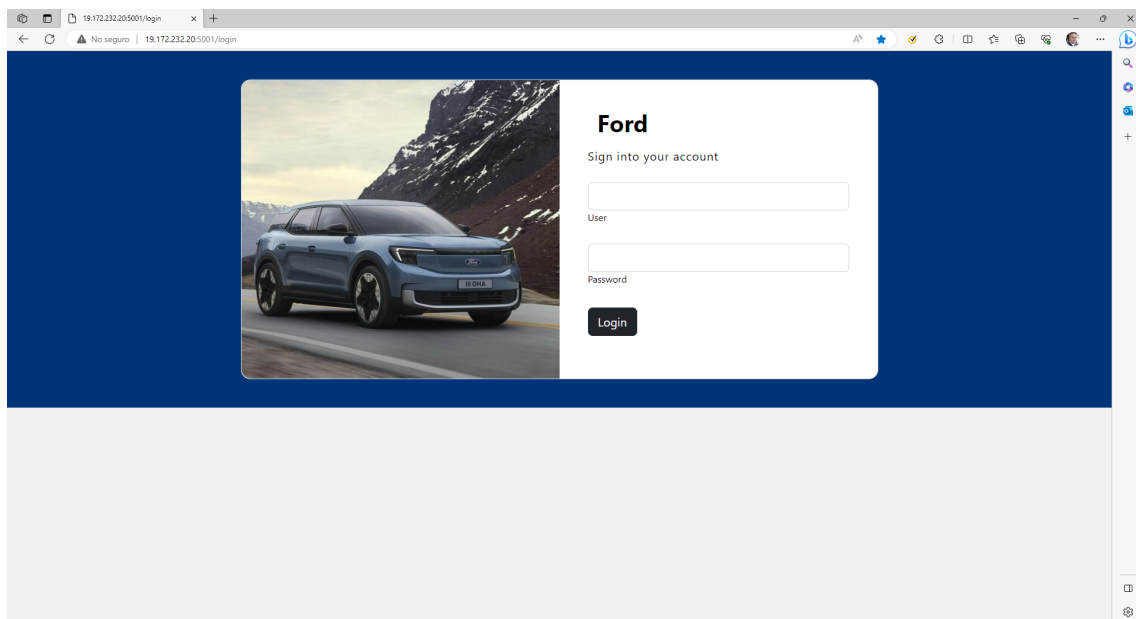


Figura 9.1: Pantalla de Inicio de Sesión

Una vez que el usuario ha iniciado sesión correctamente, accederá al panel principal de la aplicación.

9.1.2. Panel Principal

El panel principal es el centro de control desde el cual los usuarios pueden acceder a todas las funciones de la aplicación. Proporciona una visión general de la producción en curso y permite la navegación a diferentes secciones y escenarios.

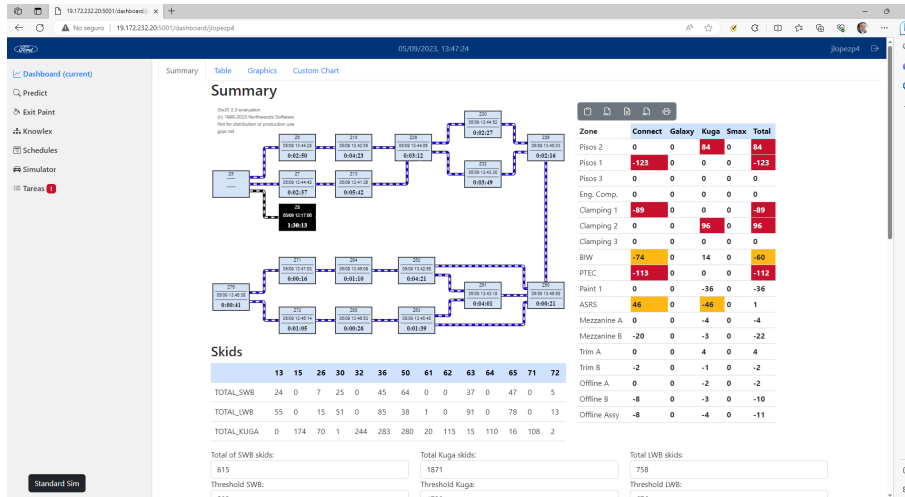


Figura 9.2: Panel Principal

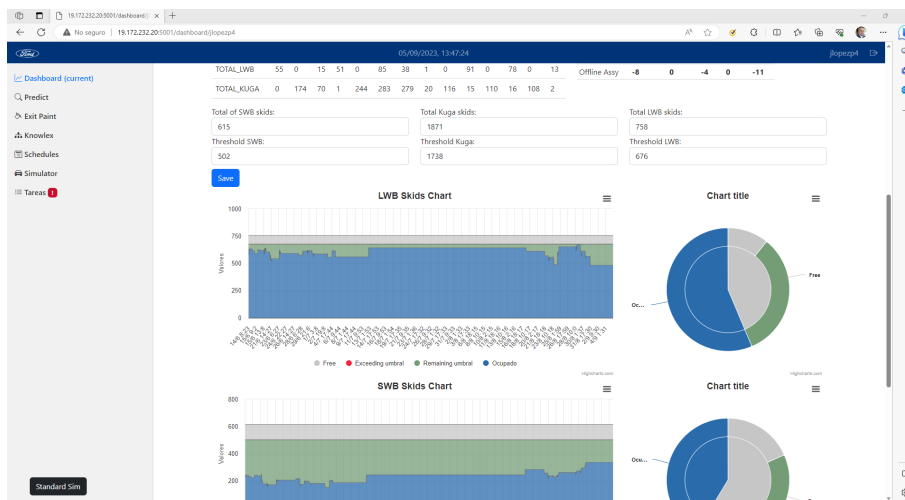


Figura 9.3: Panel Principal

9.2 Escenarios de Uso

A continuación, se describen los escenarios de uso que explicaron en la sección 4.2

9.2.1. Monitorización de Producción

Escenario

Imagina la siguiente situación en una planta de producción: eres un supervisor encargado de supervisar la producción en tiempo real y asegurarte de que se esté cumpliendo con los objetivos establecidos. Utilizas una herramienta web específica para esta tarea.

Pasos

Para llevar a cabo esta tarea, sigues estos pasos:

1. ****Inicio de Sesión****: Inicias sesión en la herramienta web utilizando tus credenciales autorizadas. Esto te da acceso al panel principal desde donde se puede supervisar la producción en tiempo real.

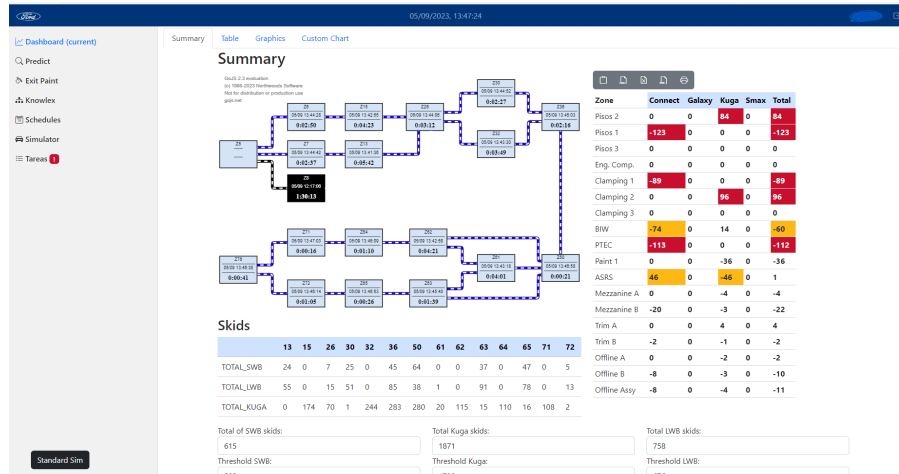


Figura 9.4: Panel Principal

Acciones

Una vez que has accedido al panel principal, puedes realizar las siguientes acciones:

- ****Monitorización en Tiempo Real****: Utilizas el panel principal para monitorizar el estado de la producción en tiempo real. Puedes visualizar áreas donde la producción se encuentra detenida o ralentizada, lo que te permite identificar posibles problemas o cuellos de botella en la línea de producción.

- ****Control Room****: Accedes al "Control Room" que se muestra en la tabla derecha del panel (ver Figura 9.4). Aquí, puedes ver el balance de producción para cada zona de la fábrica. Esto te proporciona una visión general de si la producción está en línea con los objetivos establecidos o si se están produciendo desviaciones.

- ****Toma de Medidas Correctivas****: Si identificas áreas con producción detenida o desviaciones significativas en el balance de producción, puedes tomar medidas correctivas de inmediato. Esto puede incluir la asignación de recursos adicionales a una zona con problemas o la reprogramación de la producción para optimizar los resultados.

9.2.2. Simulación de Producción

Escenario

Imagina la siguiente situación en una planta de producción de automóviles: se ha producido un fallo en una zona específica de la fábrica que está afectando la producción. Como analista de producción, tu tarea es ajustar la producción en tiempo real para minimizar el impacto del fallo y mantener el rendimiento general de la fábrica en línea con los objetivos.

Pasos

Para abordar este desafío, sigues estos pasos:

1. ****Acceso al Panel de Simulación****: Desde el panel de control principal, seleccionas la opción "Simulator". Esto te da acceso a las herramientas de simulación de producción. [9.5](#)

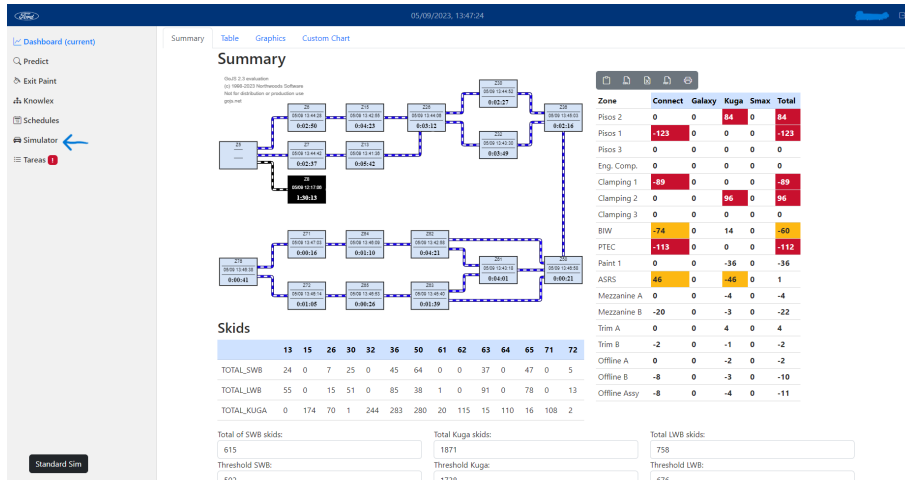


Figura 9.5: Panel de Simulación

2. ****Selección de Escenario****: Accedes a un esquema detallado de la fábrica que muestra la distribución de la producción por zonas. Activas el modo de simulación para comenzar a realizar ajustes en la producción. [9.6](#)

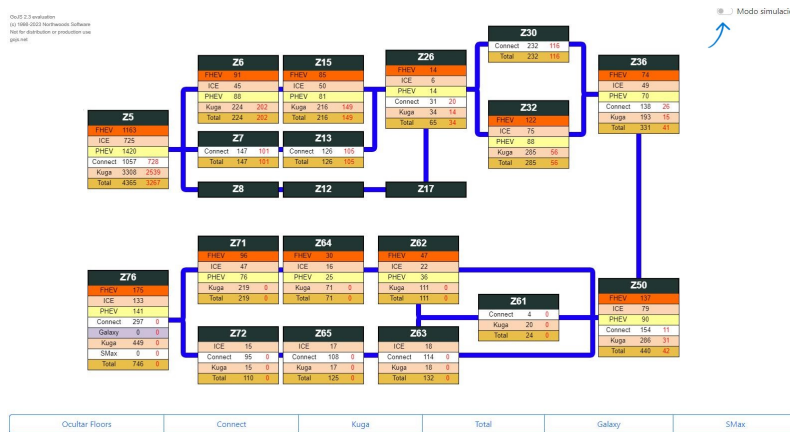


Figura 9.6: Esquema de la Fábrica

3. ****Configuración de la Simulación****: Configuras la simulación según sea necesario. Esto implica ajustar la cantidad de producción en cada zona afectada por el fallo y redistribuir los recursos para minimizar las interrupciones en la línea de producción. [9.7](#)
4. ****Ejecución de la Simulación****: Una vez que has ajustado los parámetros de la simulación, presionas el botón "Simular" y observas los datos generados por la simulación. [9.8](#)

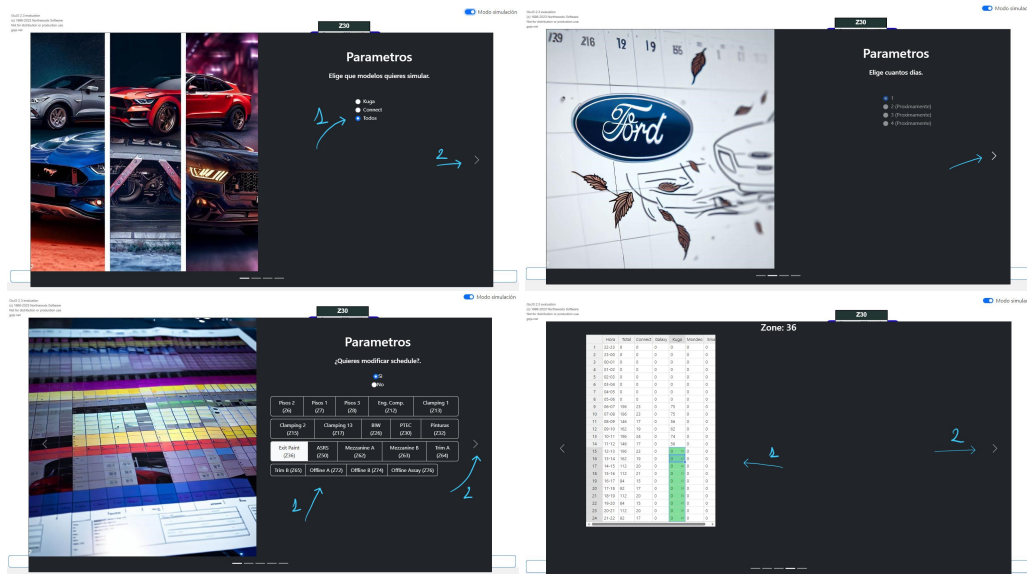


Figura 9.7: Configuración de la Simulación

Acciones

Durante la simulación de producción, tu objetivo es encontrar la configuración óptima que permita mantener el rendimiento de la fábrica en línea con los objetivos, a pesar del fallo en una zona específica. Esto implica realizar ajustes en tiempo real, como redistribuir la cantidad de coches que se hace de cada modelo, para minimizar las interrupciones y maximizar la eficiencia de la producción.

9.2.3. Predicción de Tiempos de Paso por Zona

Escenario

Supongamos que eres un supervisor o analista en una planta de producción y deseas utilizar la herramienta web para predecir los tiempos de paso por zona para un vehículo específico identificado por su número de identificación del vehículo (VIN).

Pasos

Para realizar esta predicción, sigues estos pasos:

1. ****Inicio de Sesión****: Inicias sesión en la herramienta web utilizando tus credenciales autorizadas. Esto te da acceso al panel principal.
2. ****Acceso a la Función de Predicción****: Desde el panel principal (ver Figura 9.9), navegas a la función de predicción de tiempos de paso por zona. Esta función está diseñada para ayudarte a estimar cuánto tiempo llevará que un vehículo en particular pase por cada zona de producción.
3. ****Ingreso del VIN****: Accedes a la página de predicción (ver Figura 9.10) donde puedes ingresar el número de identificación del vehículo (VIN) para el cual deseas realizar la predicción.
4. ****Ejecución de la Predicción****: Luego de ingresar el VIN, haces clic en el botón de "Ejecutar Predicción" (ver Figura 9.10) que se muestra en la página.

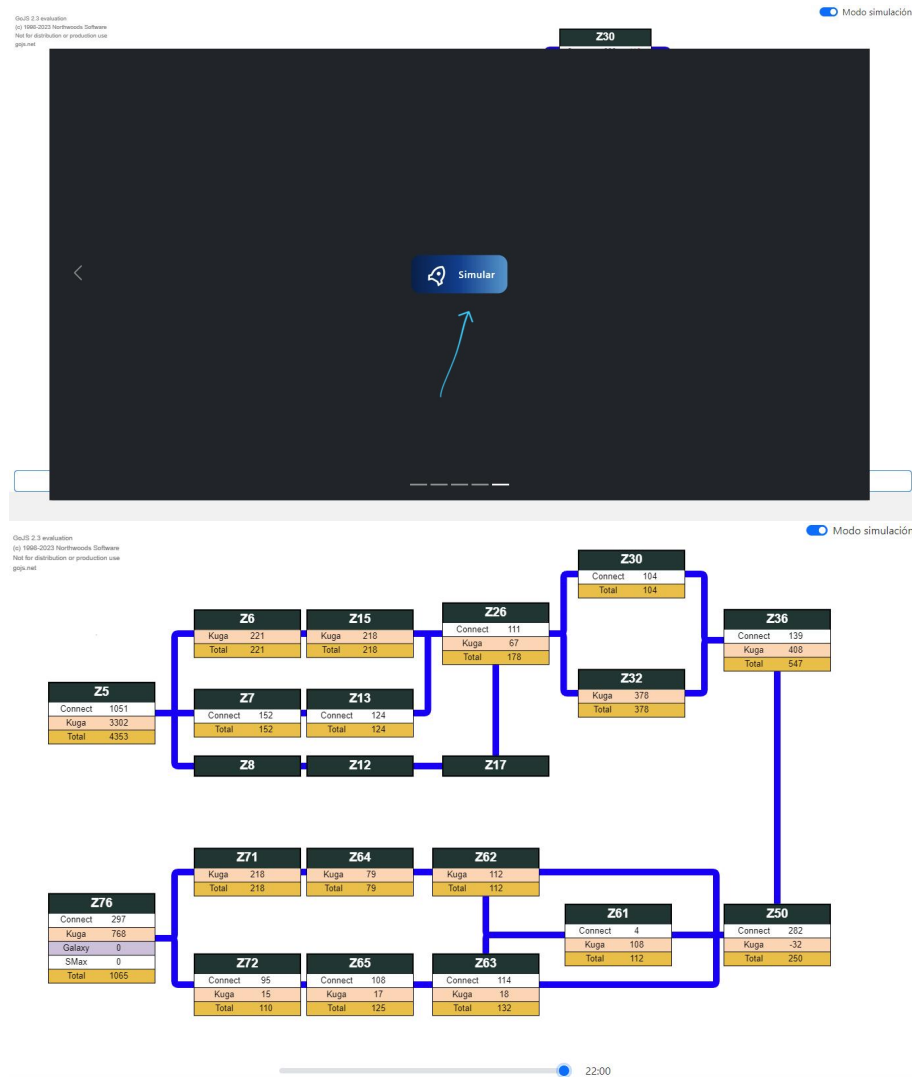


Figura 9.8: Ejecución de la Simulación

Acciones

Una vez que has accedido a la página de predicción, puedes realizar las siguientes acciones:

- ****Resultado de la Predicción****: La herramienta web procesa la solicitud y te devuelve una tabla que muestra los tiempos de paso estimados por zona para el vehículo correspondiente al VIN ingresado. Esta tabla proporciona una visión detallada de cuánto tiempo se espera que el vehículo pase en cada área de producción.

- ****Análisis y Toma de Decisiones****: Utilizas esta información para analizar el flujo de trabajo de producción previsto y tomar decisiones informadas. Por ejemplo, puedes identificar áreas donde se espera que se acumule más tiempo y tomar medidas para optimizar la eficiencia en esas zonas.

La función de predicción de tiempos de paso por zona te ayuda a planificar y gestionar la producción de manera más efectiva, permitiéndote anticipar y abordar posibles cuellos de botella y retrasos en el proceso de fabricación.

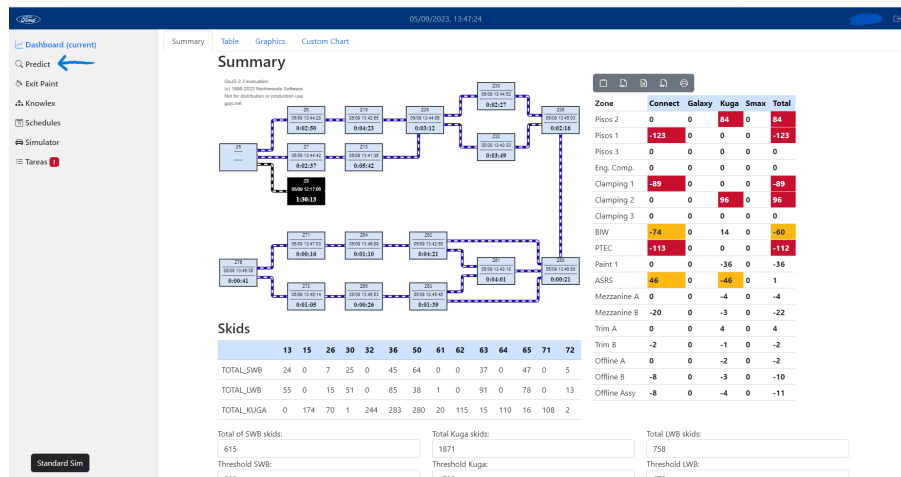


Figura 9.9: Panel Principal

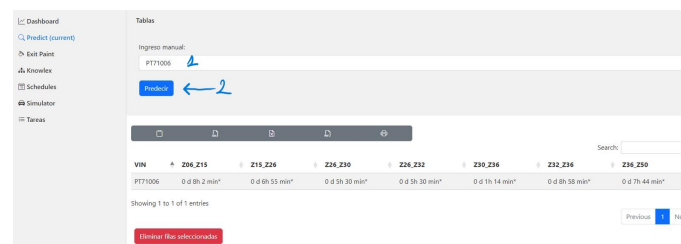


Figura 9.10: Página de Predicción

9.2.4. Análisis de Programas de Producción (Schedules)

Escenario

Supongamos que eres un analista o supervisor en una planta de producción y has iniciado sesión en la herramienta web para monitorear y analizar el rendimiento de la producción. Has notado que en el control room se registraron pérdidas significativas en una zona específica. Ahora deseas acceder a los programas de producción (schedules) para analizar cuántos coches se perdieron en esa zona en cada hora del turno.

Pasos

Para realizar este análisis, sigues estos pasos:

- **Inicio de Sesión**:** Inicias sesión en la herramienta web utilizando tus credenciales autorizadas. Esto te da acceso al panel principal.
- **Acceso a Programas de Producción (Schedules)**:** Desde el panel principal (ver Figura 9.11), navegas a la sección de "Schedules". Esta función está diseñada para proporcionar detalles sobre la programación de producción, incluido el número de coches que se deben producir y lo que se han producido realmente.

Acciones

Una vez que has accedido a la sección de "Schedules", puedes realizar las siguientes acciones:

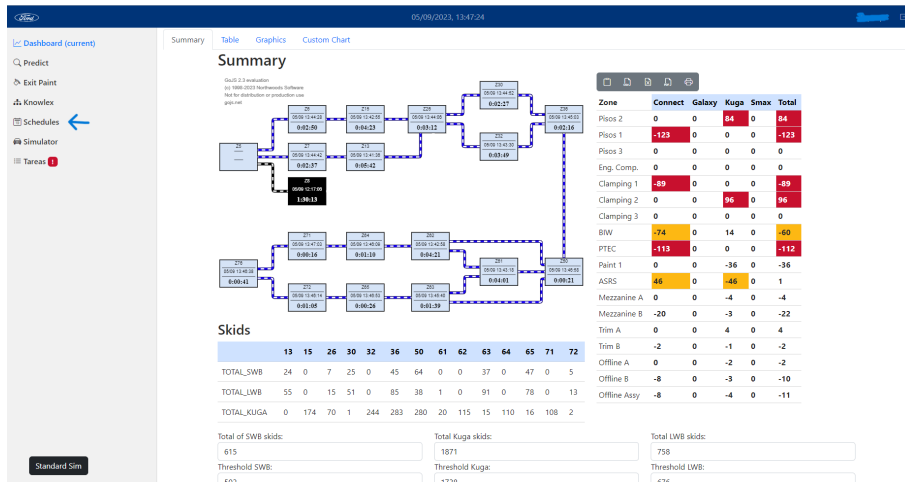


Figura 9.11: Panel Principal

- ****Selección de Zona****: Dentro de la sección de "Schedules", tienes la opción de seleccionar la zona que quieres ver su Schedule.
- ****Visualización de Datos****: La herramienta web presenta unos gráficos de cada zona que se adaptan a la pantalla para poder mostrar hasta 4 en la misma pagina para poder comparar. También se puede cambiar el modo de gráfico (barras o líneas) incluso verlo en formato tabla y exportarlo en diferentes formatos (pdf, excel, csv,etc) (ver figura 9.12).

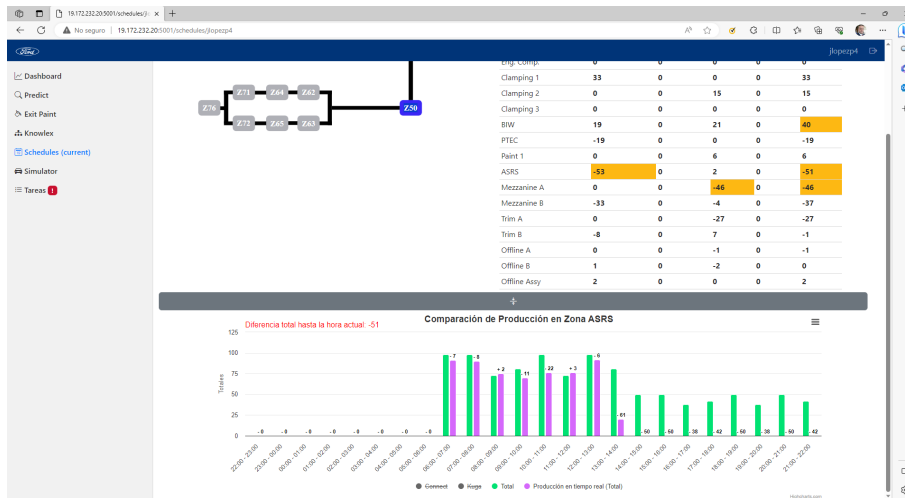


Figura 9.12: Schedules detallados

El acceso a los programas de producción (schedules) te permite realizar un seguimiento detallado de la programación de producción y analizar eventos específicos, como las pérdidas en una zona particular durante un turno de producción. Esto te brinda información valiosa para la toma de decisiones y la optimización del rendimiento de la planta de producción.

9.2.5. Creación y Seguimiento de Tareas

Escenario

Supongamos que eres un supervisor en una planta de producción y deseas asignar una tarea específica a un analista. Has iniciado sesión en la herramienta web para gestio-

nar estas tareas de manera eficiente. Tu objetivo es crear una nueva tarea, proporcionar detalles relevantes y asegurarte de que el analista designado la reciba y pueda comenzar a trabajar en ella.

Pasos

Para llevar a cabo este proceso, sigues estos pasos:

1. ****Inicio de Sesión****: Inicias sesión en la herramienta web utilizando tus credenciales autorizadas. Esto te da acceso al panel principal.

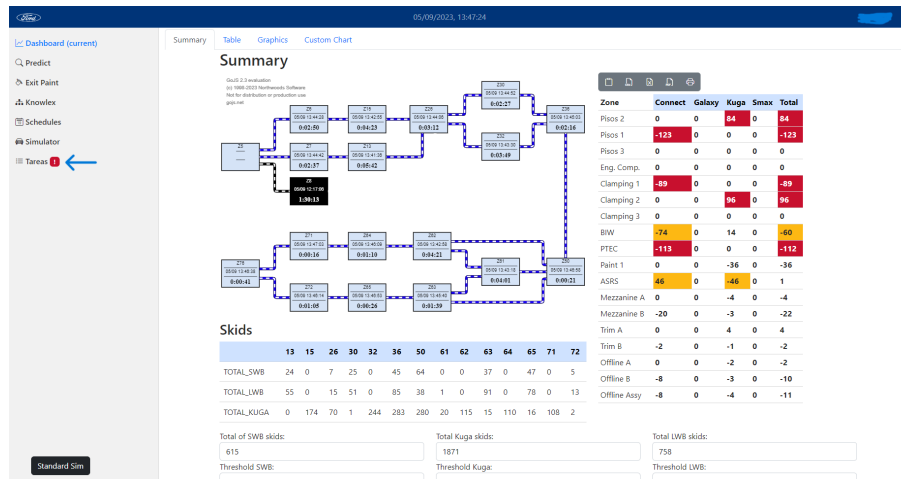


Figura 9.13: Panel Principal

2. ****Acceso a la Gestión de Tareas****: Desde el panel principal (ver Figura 9.13), navegas a la sección de "Tareas". Esta función está diseñada para la creación y seguimiento de tareas.
3. ****Creación de una Nueva Tarea****: Dentro de la sección de "Tareas", encuentras la opción para crear una nueva tarea. Al hacer clic en esta opción, se abrirá un formulario donde puedes ingresar los detalles de la tarea que deseas asignar al analista.

Acciones

Una vez que has iniciado el proceso de creación de la tarea, puedes llevar a cabo las siguientes acciones:

- ****Completar el Formulario de Tareas****: Rellenas el formulario (ver figura 9.14) de creación de tareas con la siguiente información:

- ****Título de la Tarea****: Proporcionas un título descriptivo que identifique claramente el objetivo de la tarea.
- ****Descripción****: Incluyes una descripción detallada de la tarea, especificando los requisitos, objetivos y cualquier información relevante.
- ****Fecha de Vencimiento****: Estableces una fecha límite para la tarea, lo que ayuda a gestionar el tiempo de manera efectiva.
- ****Prioridad****: Asignas una prioridad a la tarea para indicar su importancia relativa.

- ****Asignación al Analista****: Seleccionas al analista al que deseas asignar la tarea.

- ****Envío de la Tarea****: Una vez que hayas completado el formulario, envías la tarea al analista designado. Esto genera una notificación para el analista y la tarea se incluye en su lista de tareas pendientes.

Tareas de jtorresq

Tareas recibidas Tareas creadas

Título Descripción Asignada a Fecha de creación Fecha de vencimiento Progreso Prioridad Acciones

No data available in table

Agregar Tarea 1

Agregar Tarea

Título

Descripción

Creado por
jtorresq

Fecha de creación 30/08/2023 2

Fecha de vencimiento dd/mm/aaaa

Prioridad
Alta

Asignar a:
Seleccionar usuario...

Subtareas:

Ingrese cada subtarea en una línea separada.

Agregar ← 3

Figura 9.14: Panel Principal

- ****Seguimiento de Tareas****: Desde tu panel principal, puedes acceder a la sección de "Tareas" para ver el progreso de la tarea asignada. Esto incluye detalles sobre el estado actual de la tarea y su fecha de vencimiento.

- ****Comunicación con el Analista****: En caso de que sea necesario, puedes acceder a un chat privado con el analista asignado para discutir detalles adicionales, proporcionar orientación o aclarar dudas relacionadas con la tarea. (ver figura 9.15)

test

Descripción test

Creada por jlopezp4

Fecha de creación 2023-09-06

Fecha límite 2023-09-17

Prioridad Alta

Estado de las subtareas:

Chat:

¡Hola esto es una prueba

Escribe un mensaje...

Figura 9.15: Detalles Tarea

La creación y seguimiento de tareas dentro de la herramienta web te permite asignar responsabilidades de manera eficiente, asegurándote de que las tareas se completen de manera oportuna y que haya una comunicación efectiva con el analista designado.

CAPÍTULO 10

Validación

La validación heurística se convirtió en un pilar fundamental para evaluar la usabilidad de la interfaz de usuario. Dada la limitación de recursos y tiempo para llevar a cabo pruebas exhaustivas con usuarios reales, esta metodología ofreció una forma eficaz de identificar problemas potenciales de usabilidad en las etapas tempranas de diseño.

Como diseñador de interfaces, adopté un enfoque basado en heurísticas de usabilidad ampliamente reconocidas y respaldadas por la investigación y la experiencia en el campo. Algunas de las fuentes que consulté para definir estas heurísticas incluyeron:

- **“Design of Everyday Things” de Don Norman:** Este influyente libro destaca principios clave de diseño centrado en el usuario, como la visibilidad del estado del sistema y la retroalimentación del usuario.
- **“Usability Engineering” de Jakob Nielsen:** Nielsen es un referente en el campo de la usabilidad y su libro proporciona una sólida base en evaluación heurística.
- **“10 Usability Heuristics for User Interface Design” de Jakob Nielsen:** En este documento, Nielsen presenta sus diez heurísticas de usabilidad ampliamente aceptadas.
- **Páginas web de referencia como Usability.gov y Nielsen Norman Group:** Estas fuentes en línea ofrecen una riqueza de información sobre heurísticas de usabilidad y mejores prácticas en diseño de interfaces.

Durante la aplicación de la validación heurística, revisé nuestra interfaz en busca de problemas comunes relacionados con la usabilidad. Algunos de los criterios de evaluación que utilicé incluyeron:

1. **Visibilidad del estado del sistema:** Me aseguré de que los usuarios siempre supieran en qué parte de la aplicación se encontraban y qué acciones estaban disponibles.
2. **Consistencia y estándares:** Verifiqué que mantuviéramos una interfaz coherente en términos de diseño y terminología, siguiendo estándares de la industria cuando fuera relevante.
3. **Control y libertad del usuario:** Confirmé que los usuarios podían deshacer acciones no deseadas y navegar de manera eficiente por la aplicación.
4. **Prevención de errores:** Implementé medidas para prevenir errores comunes y garantizar que los usuarios recibieran retroalimentación adecuada si cometían errores.

5. **Reconocimiento en lugar de recuerdo:** Diseñé la interfaz para que la información importante estuviera siempre visible y no requiriera que los usuarios recordaran detalles de una pantalla a otra.

La validación heurística resultó ser un proceso valioso para mejorar la usabilidad de nuestra interfaz. Identificamos y abordamos problemas antes de la implementación final, lo que nos permitió ofrecer una experiencia de usuario más sólida y coherente. Además, esta metodología nos brindó un marco sólido respaldado por la investigación y la experiencia, lo que aumentó la confianza en la calidad de nuestro diseño.

En nuestro enfoque de validación heurística, también realizamos un análisis detallado de la interfaz de usuario de la aplicación. A continuación, presento un análisis de la página principal de la aplicación, que es un componente crucial de la interfaz y sirve como punto de entrada para los usuarios.

10.0.1. Análisis de la Página Principal

La página principal (9.2) de la aplicación se diseñó con un enfoque en la usabilidad y la presentación efectiva de información clave. A continuación, describo las observaciones clave:

1. **Diseño Visual Atractivo:** La página principal presenta un fondo azul que contribuye a una apariencia profesional y coherente con la identidad de la marca Ford. La elección de colores como el rojo y el negro para resaltar información importante agrega impacto visual y ayuda a los usuarios a identificar áreas críticas.
2. **Uso de Gráficos:** La página principal incluye varios gráficos que proporcionan a los usuarios una visión general de la situación en la fábrica. Estos gráficos permiten a los usuarios identificar tendencias y patrones clave de producción de manera rápida y eficiente.
3. **Indicador de Última Actualización:** La inclusión de la hora de la última consulta a la base de datos es una característica valiosa para los usuarios. Esto les brinda confianza en la actualidad de los datos que están viendo, lo que es especialmente importante en entornos donde la precisión de los datos es esencial.
4. **Indicador de Carga:** Durante las operaciones que requieren tiempo de procesamiento, como la carga de datos, se muestra una animación del logo de Ford. Este indicador de carga proporciona retroalimentación visual al usuario, indicando que su solicitud se está procesando. Esta técnica mejora la experiencia del usuario al reducir la incertidumbre y la ansiedad.
5. **Organización Lógica:** Los elementos en la página principal están organizados de manera lógica. La información clave, como los gráficos y los indicadores de estado, se presenta de manera clara y ordenada. Esto facilita a los usuarios encontrar la información que necesitan de manera eficiente.

10.0.2. Conclusiones

La web se diseñó cuidadosamente para ofrecer una experiencia de usuario efectiva. Los principios de usabilidad y diseño centrado en el usuario se reflejan en la disposición de elementos, el uso de colores y gráficos, y las características que brindan orientación y retroalimentación al usuario.

CAPÍTULO 11

Conclusions

En este Trabajo de Fin de Grado, se ha abordado de manera integral el desarrollo de una herramienta web destinada a mejorar la gestión y optimización de procesos en Ford Motors. A lo largo del proyecto, se ha aplicado un enfoque iterativo que ha demostrado ser altamente efectivo en la consecución de los objetivos planteados y en la creación de un producto final que cumple con las expectativas y necesidades de los usuarios y stakeholders involucrados.

El enfoque iterativo adoptado ha permitido un proceso de diseño, implementación y evaluación continuo, lo que a su vez ha posibilitado la detección temprana de problemas y la introducción de mejoras de manera ágil. La colaboración constante con el supervisor y la incorporación de su retroalimentación han resultado fundamentales en la evolución de la herramienta, asegurando su alineación con los requisitos y objetivos del proyecto.

La comparación de diferentes enfoques de desarrollo de software evidenció las ventajas distintivas del método iterativo en términos de flexibilidad, adaptabilidad y mitigación de riesgos. Esta elección metodológica se basó en la naturaleza evolutiva del proyecto y en la necesidad de crear una herramienta que pudiera adaptarse a las cambiantes demandas del entorno automovilístico.

Los resultados obtenidos corroboran la efectividad del desarrollo iterativo en la construcción de sistemas complejos y funcionales. La herramienta web desarrollada no solo ha satisfecho los requerimientos iniciales, sino que ha logrado superar las expectativas al ofrecer respuestas un 90 % más rápidas y la incorporación de nuevas herramientas que aun a día de hoy usan diariamente.

En última instancia, este proyecto ha demostrado la importancia de un enfoque metodológico adecuado en el proceso de desarrollo de software. La elección del desarrollo iterativo ha sido respaldada por su capacidad para adaptarse a situaciones cambiantes, su enfoque en la retroalimentación constante y su habilidad para mitigar riesgos. El conocimiento y las habilidades adquiridas a lo largo de este proyecto no solo son valiosos en el contexto académico, sino que también son transferibles y aplicables en futuros proyectos de desarrollo de software en la industria automotriz y en el mundo tan cambiante de la logística.

En resumen, este Trabajo de Fin de Grado ha sido una oportunidad enriquecedora para explorar y aplicar conceptos teóricos en un entorno práctico y real. El éxito de la herramienta web desarrollada valida la elección del enfoque iterativo y subraya la importancia de una colaboración efectiva y la adaptación constante en el proceso de desarrollo de software.

Bibliografía

- [1] Ford Motor Company. (n.d.). - Wikipedia. Consultado en https://es.wikipedia.org/wiki/Ford_Motor_Company
- [2] Ford Argentina. (n.d.). - Wikipedia. Consultado en https://es.wikipedia.org/wiki/Ford_Argentina
- [3] Henry Ford. (n.d.). - Wikipedia. Consultado en https://es.wikipedia.org/wiki/Henry_Ford
- [4] Desarrollo iterativo y creciente - Wikipedia, la enciclopedia libre. Consultado en https://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente.
- [5] Metodologías de Desarrollo de Software: ¿Cuál Elegir? - Innevo. Consultado en <https://www.innevo.es/metodologias-de-desarrollo-de-software-cual-elegir/>.
- [6] Cuadro comparativo de metodologías de software. - cuadrocomparativode.net. Consultado en <https://cuadrocomparativode.net/cuadro-comparativo-de-metodologias-de-software/>.
- [7] Heuristic evaluation - Wikipedia, la enciclopedia libre. Consultado en https://en.wikipedia.org/wiki/Heuristic_evaluation.
- [8] HATEOAS: ¿cuál es el principio que oculta este acrónimo? - Ionos Consultado en <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/hateoas-que-es-y-cual-es-su-funcion-en-las-api-rest/>.

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

La contribución de mi Trabajo de Fin de Grado (TFG) a los Objetivos de Desarrollo Sostenible (ODS) es un aspecto fundamental que merece un análisis detallado. Mi proyecto se relaciona principalmente con tres ODS clave, a saber: el ODS 8, el ODS 9 y el ODS 11. A continuación, profundizaré en cómo mi trabajo impacta en cada uno de estos objetivos y cómo, en última instancia, contribuye a un futuro más sostenible y equitativo.

1. ODS 8: Trabajo decente y crecimiento económico

Mi TFG se alinea estrechamente con el ODS 8, que se centra en la promoción del trabajo decente y el crecimiento económico sostenible. Al desarrollar una herramienta de gestión de producción, he abordado directamente la cuestión del trabajo decente en la industria manufacturera. Mi proyecto busca aumentar la eficiencia en la producción, lo que, a su vez, puede generar un aumento en la demanda de mano de obra calificada y no calificada.

El incremento de la eficiencia en la producción tiene el potencial de crear más empleos, especialmente en las regiones donde se implemente esta herramienta. En un mundo donde el desempleo y la subocupación son desafíos persistentes, mi proyecto puede contribuir significativamente al ODS 8 al promover oportunidades de trabajo decente y, en última instancia, al impulsar el crecimiento económico sostenible.

2. ODS 9: Industria, innovación e infraestructuras

El ODS 9 se centra en el desarrollo de industrias, la promoción de la innovación y la construcción de infraestructuras resilientes. Mi proyecto se alinea perfectamente con estos objetivos, ya que la herramienta de gestión de producción que he desarrollado incorpora elementos de innovación en la gestión de procesos industriales.

Al introducir características como la simulación y la predicción en la gestión de la producción, mi proyecto busca mejorar significativamente la eficiencia en la industria manufacturera. Estas innovaciones no solo optimizan los procesos existentes, sino que también permiten una mayor flexibilidad y adaptación a los cambios en la demanda del mercado. Esto es esencial para el crecimiento económico sostenible y la competitividad de la industria.

Además, el desarrollo de esta herramienta también puede considerarse una infraestructura digital que las empresas manufactureras pueden adoptar para mejorar sus operaciones. Esto está en línea con el ODS 9, que busca promover la construcción de infraestructuras resilientes y la adopción de tecnologías avanzadas.

3. ODS 11: Ciudades y comunidades sostenibles

Aunque mi proyecto se centra en la gestión de producción en entornos industriales, también tiene un impacto indirecto en la sostenibilidad de las comunidades. La eficiencia mejorada en la producción industrial puede reducir el desperdicio de recursos naturales y energía, lo que es esencial para lograr comunidades más sostenibles.

Además, la reducción de los impactos negativos en el medio ambiente asociados con la producción industrial, como la contaminación y la generación de residuos, también contribuye a la creación de comunidades más limpias y saludables. Estos son aspectos clave del ODS 11, que busca construir ciudades y comunidades sostenibles.

En resumen, mi proyecto de TFG no solo se alinea con los ODS 8, 9 y 11, sino que también tiene el potencial de impactar positivamente en otros ODS relacionados, como la reducción de la pobreza y la igualdad de género, al impulsar mejoras en la eficiencia económica y la creación de empleo. A través de la innovación en la gestión de la producción y la promoción de la sostenibilidad, mi proyecto es un paso hacia un futuro más sostenible y equitativo.