

PÁGINAS WEB PARA LA EMPRESA “YYCD PERFORMANCE” EN GITHUB

Fernando Carillo, Daniela Murillo, Andy Rojas, David Gonzales

Universidad Fidélitas

Heredia, Costa Rica

fcarrillo80174@ufide.ac.cr

amurillo40573@ufide.ac.cr

arojas60114@ufide.ac.cr

egonzalez80628@ufide.ac.cr

Resumen- El presente documento busca explicar la realización del proyecto denominado como “YYCD Performance” utilizando la plataforma GitHub en su elaboración y como se desarrolló la página web que funciona como proyecto final y deja demostrar las habilidades aprendidas durante su elaboración.

Palabras clave: GitHub, Repositorio, Java, Web, Localhost, programación, Tunning.

I. INTRODUCCIÓN

La plataforma de GitHub es una plataforma en la nube de internet que cumple con el rol de servir de almacenamiento de código fuente para distintos lenguajes de programación (especialmente Java) y de archivos de un proyecto que abarque temas informáticos como software, web, aplicaciones móviles, entre otros. Esta herramienta le permite a programadores llevar un control de distintas versiones o archivos de tu proyecto ya que esta permite crear distintas versiones de un mismo proyecto que se almacena en un repositorio y así permitir trabajar en conjunto, ya que GitHub no solo funciona para que un trabajador guarde su proyecto desde la nube sino que también ayuda a otros trabajadores en su mayoría compañeros de programación que estén trabajando en un mismo proyecto se puedan conectar y visualizar y editar el mismo trabajo al igual que actualizar de manera sincrónica sus correcciones. GitHub al ser de código abierto permite realizar proyectos compartidos y mantener el seguimiento detallado de su progreso y además funciona como red social conectando a los desarrolladores con los usuarios.

GitHub fue creada por los estadounidenses Chris Wanstrath, PJ Hyett, Tom Preston-Werner y Scott Chacon en Febrero del 2008. Por medio de una entrevista que los integrantes realizaron en Yahoo dieron a conocer que su creación ya contaba un año después con un total de 46 mil repositorios públicos que daba una idea de lo popular que se volvería su plataforma en el mundo de la programación. Hasta el día de hoy GitHub sigue creciendo constantemente y se ha vuelto la plataforma preferida entre montones para guardar proyectos en la nube que incluso hasta fecha con competencias de renombre como Bit Bucket, Mercurial y GitLab sigue dominando. [

El presente proyecto busca demostrar como a través de la programación en Maven Spring Boot Inializr Project en Java

fue posible la creación de una página web cuya finalidad hacia el usuario es dar a conocer distintas ofertas de trabajo mecánicas que pueden darle a su vehículo por medio de una práctica conocida popularmente como “Tunning”. Esta hace que los usuarios que tengan un medio de transporte (en este caso un auto) puedan realizarle distintas modificaciones internas y externas a sus vehículos de manera personalizada y puedan consultar los precios al crearse una cuenta en la página que los deja acceder a todo lo que la compañía le ofrece.

A. Justificación

En un entorno digital, el uso de herramientas colaborativas para el desarrollo de software ha sido de uso constante por programadores y trabajadores de la informática en muchos aspectos se ha convertido en una práctica esencial. El presente proyecto surge con el objetivo de demostrar el conocimiento adquirido en el desarrollo web y la programación orientada a servicios, mediante la creación de una plataforma digital simulando el papel de una compañía llamada *YYCD Performance*. Esta plataforma ofrece servicios de personalización vehicular, conocidos como “Tuning”, permitiendo a los usuarios explorar opciones de modificación estética y funcional para sus automóviles. GitHub fue elegido como herramienta principal de control de versiones y colaboración, no solo por su facilidad de uso, sino también por su capacidad de integrarse con flujos de trabajo ágiles, permitiendo la gestión estructurada del proyecto desde sus primeras etapas hasta el producto final. De esta forma, se pone en evidencia la importancia de dominar plataformas como GitHub para el desarrollo de proyectos reales, colaborativos y escalables.

B. Objetivos

1) *Objetivo general.*

Nuestro objetivo es desarrollar una plataforma web para la empresa ficticia *YYCD Performance* utilizando tecnologías como Java y Spring Boot, integrando GitHub como sistema de control de versiones para evidenciar el uso de buenas prácticas en la gestión de proyectos de software colaborativos.

2) *Objetivos específicos.*

- Implementar una arquitectura web basada en Java y Spring Boot que permita la visualización de los servicios de tuning ofrecidos por la empresa.
- Utilizar GitHub como repositorio central para almacenar y gestionar el código fuente del proyecto, manteniendo un control de versiones eficiente y colaborativo.
- Diseñar una interfaz de usuario que permita a los clientes consultar información, cotizar servicios y registrarse para acceder a funciones exclusivas.

II. DESARROLLO DE CONTENIDOS

C. Programación de paquetes

El proyecto cuenta con una serie de paquetes que cumplen distintos roles, cada uno cuenta con una o varias clases que a su vez tiene código que ayuda en la creación de la página. A continuación vamos a explicar que realiza cada paquete y en que se basa el código de sus clases.

1) Paquete YYCD.YYCD: Este tiene dos clases de java con los nombres “ProjectConfig” y “YYCDPerformanceApplication” Estas iniciarán la aplicación de Spring Boot, funcionando como un método estático lo que harán será arrancar el entorno de Spring Boot con el servidor web embebido, la configuración de dependencias y el escaneo de comando como “@Component, @Service, @Controller” entre otros.

2) Paquete YYCD_YYCD.controller: Este paquete se divide en dos clases: “CotizarController” y “ServiciosController” lo que estas clases ejecutan es, en caso de la de cotizar, hace peticiones web que se relacionen con la cotización de servicios de la empresa, mientras que la de servicios muestra los servicios disponibles dentro de la plataforma web.

3) Paquete YYCD_YYCD.dao: El paquete dao cuenta únicamente con una clase llamada “ServicioDao” que define una interfaz de acceso a datos para la clase y se encarga de interactuar con la base de datos.

4) Paquete YYCD_YYCD.domain: Este paquete tiene un total de cuatro clases denominadas: “Contacto” “Cotización” “Review” y “Servicio”. Las clases están diseñadas para formar parte del modelo de datos de tu aplicación por lo que cada una juega el papel de tabla donde se guardan los datos relacionados con la actividad de la empresa (dependiendo de cual se elija)

5) Paquete YYCD_YYCD.impl: Esta conforma la lógica de implementación del sistema que implementa interfaces de la capa de servicios que definen como se comportan las operaciones de negocio. Las dos clases con las que cuentan son “CotizacionServiceImpl” y “ServicioServiceImpl” que la primera simula la creación de cotizaciones con IDs únicos y la segunda accede a la base de datos para listar los servicios.

6) Paquete YYCD_YYCD.service: Las dos clases que forman parte de este paquete llamadas “CotizacionService” y “ServicioService” forman parte de la capa de servicios de la aplicación actuando como puente entre los controladores y la lógica de negocio. Por una parte se crea una cotización y por la otra se obtienen todos los servicios disponibles.

D. Programación de application.properties

El archivo application.properties funciona para configurar comportamiento del sistema (servidor web, conexión con SQL). El código en esta parte define el nombre de la aplicación, el servidor web, muestra la hora en los logs del sistema, se configura la ubicación el tipo y codificación de las vistas de html, define como se conecta a la base de datos (en este caso SQL) y finalmente que las vistas se carguen bien desde Thymeleaf.

```
1  spring.application.name=YYCD_Performance
2
3  server.port=80
4
5  logging.pattern.dateformat=hh:mm
6
7  spring.main.banner-mode=off
8
9  spring.thymeleaf.cache=false
10
11  spring.datasource.url=jdbc:mysql://localhost:3306/taller_db
12  spring.datasource.username=usuario_prueba
13  spring.datasource.password=Usuario_Clave.
14  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
15  spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
16
17  spring.jpa.properties.hibernate.format_sql=true
18  logging.level.org.hibernate.SQL=DEBUG
19  logging.level.org.type.descriptor.sql.BasicBinder=TRACE
20  spring.thymeleaf.prefix=classpath:/templates/
21  spring.thymeleaf.suffix=.html
22  spring.thymeleaf.mode=HTML
23  spring.thymeleaf.encoding=UTF-8
24  spring.thymeleaf.servlet.content-type=text/html
25
```

Figura 1: Código de application.properties

E. Programación de messages.properties

Los dos archivos con los nombres “messages.properties” contienen mensajes con claves o valores como por ejemplo tipos de palabras clave de login (contraseña, errores, username) también como acciones y errores que pueden aparecer durante su ejecución.

```
1  login.login=Login
2  login.username=Username
3  login.password=Password
4  accion.registrar=Register
5  accion.recordar=Remember
6  error.login=Login error
```

Figura 2: Código dentro del messages.properties

```
1  login.login=Iniciar Sesión
2  login.username=Usuario
3  login.password=Contraseña
4  accion.registrar=Registrarse
5  accion.recordar=Recordar
6  error.login=Error en el inicio de sesión
```

Figura 3: Código dentro de messages_es.properties

F. Programación de static (HTML)

1) Login.html: Este archivo presenta un formulario de inicio de sesión para que los usuarios ingresen su nombre de usuario y contraseña.

2) Apariencia.html: Esta página está diseñada para mostrar los servicios relacionados con la personalización estética de vehículos.

3) Contacto.html: Proporciona información de contacto clara y accesible. Desde mostrar la ubicación exacta de la empresa, hasta presentar los miembros del equipo y su función que mantiene el mismo diseño visual del resto del sitio para coherencia de marca.

4) Cotizar.html: Esta página permite que un usuario solicite la cotización que desea realizarle a su vehículo.

5) Equipamiento.html: Esta página muestra el servicio de equipamiento con la que cuenta la empresa para realizar las modificaciones necesarias al vehículo.

6) Iluminación.html: Esta página muestra los productos de iluminación que puedes agregarle a la modificación de tu auto.

7) Index.html: Este archivo nos manda a la página principal del sitio web donde se encuentran las otras opciones que pueden seleccionar en la página si está interesado en seguir investigando.

8) Nosotros.html: Este apartado contiene un poco de historia sobre porque y como se configuró la página.

9) Review.html: Este apartado se configuró para que muestre las reseñas del sitio web sobre el servicio que han recibido durante su experiencia con la compañía (sea buena o mala).

10) Servicios.html: Este apartado muestra los servicios que ofrece la empresa a sus compradores dando múltiples opciones dependiendo de lo que el cliente desea realizarle a su vehículo.

III. CONCLUSIONES

Al ejecutar el código se mostraría lo siguiente:

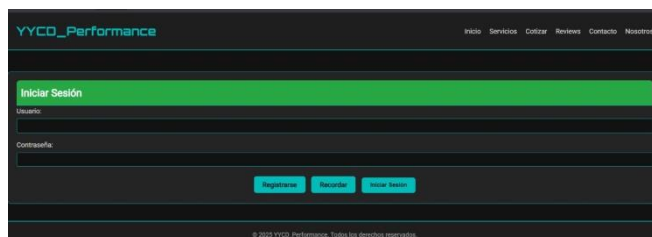


Figura 4: Página de Login



Figura 5: Página de Servicios

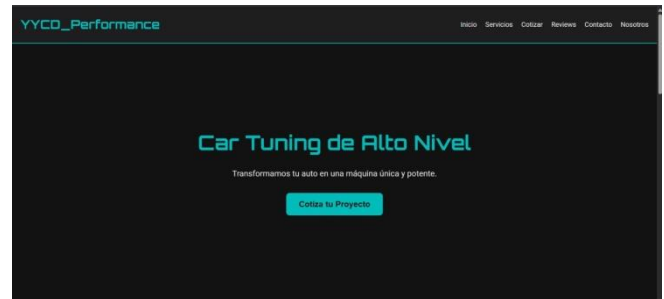


Figura 6: Página de Index

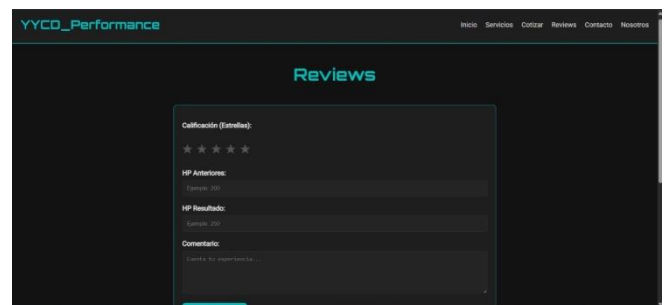


Figura 7: Página de Review



Figura 8: Página de Cotización

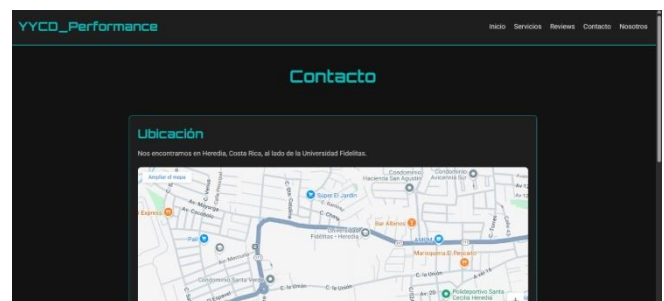


Figura 9: Página de Contacto



Figura 10: Página de Nosotros

A. Recomendaciones

1) Se recomienda para próximos proyectos lograr que la página logre ingresar desde cualquiera de los botones de html sin necesidad de estar en el index principal. Ya que de ser que se este realizando alguna acción en la página y cambie de opinión y busque otro botón más este nos dará error.

2) Las imágenes utilizadas en el proyecto por los distintos tamaños de las mismas hacían que a la hora de cargar el proyecto mostraran dificultades para cargar las imágenes y que se adaptaran al estilo de la página.

D. Referencia bibliográfica

“Qué es Github y para qué sirve: una guía para principiantes”. Ebac. Accedido el 23 de abril de 2025. [En línea]. Disponible: <https://ebac.mx/blog/que-es-github> [1]

“Una breve historia de Github”. CANVAS, SWOT, PESTEL & BCG Matrix Editable Templates for Startups. Accedido el 23 de abril de 2025. [En línea]. Disponible: <https://canvasbusinessmodel.com/es/blogs/brief-history/github-brief-history?srsId=AfmBOoplSpktY21OvSCw2t6mzUJpB5htt w8B3BCBs2SHEUWPbD0TsIf>. [2]

“Acerca de GitHub y Git - Documentación de GitHub”. GitHub Docs. Accedido el 23 de abril de 2025. [En línea]. Disponible: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git> [3]

“Configuración de un repositorio - Azure DevOps”. Microsoft Learn: Build skills that open doors in your career. Accedido el 23 de abril de 2025. [En línea]. Disponible: <https://learn.microsoft.com/es-es/devops/develop/git/set-up-a-git-repository> [4]

[
{
}]