

# VINTAGE CARS

---

Son muchos los motivos por los que puedes necesitar un coche antiguo de alquiler: una escapada romántica, un concierto con amigos, para una fecha especial como tú casamiento. Sea cual sea tu necesidad, en Vintage cars te ofrecemos el alquiler de vehículos con el que podrás disfrutar de un coche adaptado a tus necesidades en todo momento. Decir que nuestros vintage cars es un sistema de alquiler de vehículos no es suficiente para describir lo que realmente ofrece este programa. Se trata de un servicio de calidad que se tramita de manera muy sencilla y que, además, te permite disfrutar de manera asequible de la opción que más se adapta a tus necesidades.

Con nuestro sistema podrás elegir el vehículo que más se adapta a tus necesidades, podrás elegir entre nuestros vehículos situados en diferentes ciudades, y con distintas fechas disponibles.

También podrás seleccionar un vehículo por categoría ya sea Clásicos Europeos, Clásicos Americanos, Camionetas y mucho más.

Una vez seleccionado el vehículo de tu preferencia podrás acceder a él para ver su localización en el mapa, diferentes fotos del interior y exterior del vehículo, políticas de la página y todos los datos que necesitas saber. Si este es tu vehículo de tus sueños para esta ocasión podrás acceder a reservar tu vehículo para lo cual te pediremos que estés logueado.

Una vez realizada la reserva ya podrás disfrutar de tu paseo de ensueño.

Si no encuentras el vehículo que estas buscando no te preocupes nuestro sistema cuenta con una pagina que permite subir fácilmente productos nuevos, con sus fotos , descripciones y políticas así que a medida que nuevos autos estén disponibles los podrás visualizar rápidamente en nuestro sistema.

También podrás visualizar tus reservas una vez este logueado.

Esperemos que esta sea una experiencia única y que disfrutes mucho de nuestro sistema.

El equipo de Vintage Cards.

Nuestro principal objetivo como grupo es desarrollar este sistema de alquiler de autos antiguos, que sea un sistema fácil para el usuario y sus necesidades así como también para el administrador , que sea intuitivo , fácil de predecir y que el cliente pueda llegar a realizar su reserva sin ningún problema o contratiempo.

Objetivo que persigue el grupo al realizar el proyecto






# Equipo

---

## Katyana Soledad Prieto:

Soy Ingeniera Química recibida en el ITBA y el ultimo año de carrera lo realice en la Politécnica de Cataluña en Barcelona, España. Trabaje 3 años y medio en Procter&Gamble Mataró, Barcelona. Luego volví al país y trabajé en L'Oreal argentina por 4 años. Cree mi propia marca de esmaltes con una socia JUKA la cual vendimos este año a principio de año y empecé a estudiar la carrera de Certificate Tech Development.






En los diferentes Spring realice tareas tanto de back end, Testing y de Front End pero más de este último.

TECNOLOGIA	FORTALEZA
FRONT END	
BACK END	
INFRAESTRUCTURA	
TESTING/QA	
BASE DE DATOS	

### Joaquin Marmol:

Tengo 20 años, vivo en Necochea, provincia de Buenos Aires, aún no he tenido experiencia laboral, he terminado el secundario y durante el año 2021 estudié una carrera en la facultad de economía en Mar del Plata que luego de ese primer año deje y decidí cambiarme de rama y probar con la programación. Antes de iniciar la carrera de certified tech developer en Digital House no contaba con ningún conocimiento acerca de la programación.






Durante el proyecto integrador yo estuve enfocado en el área de la infraestructura, usando GitLab con el equipo, creando toda la red de infraestructura en la nube de Amazon Web Service, implementando pipelines de integración continua y entrega continua (CI/CD) y utilizando un bucket s3 de AWS como contenedor de imágenes para luego utilizar en nuestra base de datos proporcionada por el equipo de Digital House.

TECNOLOGIA	FORTALEZA
FRONT END	
BACK END	
INFRAESTRUCTURA	
TESTING/QA	
BASE DE DATOS	

## Nicolás Gambino

Tengo 21 años, vivo en Mendoza. Todavía no tengo experiencia laboral en el mundo IT pero este proyecto ha sido un acercamiento muy interesante. Estudié un año Licenciatura en economía y un año Ingeniera en Sistema ahí me di cuenta lo que me gustaba y empecé este largo y lindo camino.

Durante este proyecto realizado junto a mis compañeros yo me desarrollé más en la parte del Back End y Base de datos además participe en varias parte del Front donde se necesitaba mas logica y menos conocimientos técnicos de React






TECNOLOGIA	FORTALEZA
FRONT END	
BACK END	
INFRAESTRUCTURA	
TESTING/QA	
BASE DE DATOS	

### Karen Céspedes Rivas:

Me encuentro finalizando la carrera de Arquitectura en la Universidad de Buenos Aires, lo cual me dio la experiencia sobre el diseño y modelado 3d. Decidí incursionar en lo que es la tecnología de las impresiones 3d y me enfoco en el diseño 3d en Ceri3D, el cual es un emprendimiento propio que tengo desde el año 2020.

Antes de iniciar la carrera de CTD, tenía conocimientos sobre Fundamentos de la programación, adquirida por un curso impartido por la UTN.






En los diferentes Sprint fui rotando según las tareas que tenía entre Backend, Frontend y Base de Datos, también generé parte del diseño en Figma para tener una guía.

TECNOLOGIA	FORTALEZA
FRONT END	
BACK END	
INFRAESTRUCTURA	
TESTING/QA	
BASE DE DATOS	

## Fernando Maximiliano Pajón

Soy de la ciudad de Córdoba y trabajo en una empresa panificadora de mi zona. Aunque estudié carreras relacionadas con las ciencias sociales tales como derecho, relaciones públicas y periodismo (Universidad Nacional de Córdoba) siempre me sentí muy atraído por la tecnología y el desarrollo, aunque nunca incursioné en esta área hasta que en el año 2021, Digital House me permitió estudiar esta carrera y hoy me encuentro terminando el primer año con este increíble y desafiante proyecto integrador.






Durante este trayecto, me desempeñé principalmente en el área de testing, realizando todas las tareas tendientes a otorgar la mayor calidad posible al producto, y reducir los bugs o defectos que pudiera tener. También realicé algunas tareas de frontend (jest y testing library)

TECNOLOGIA	FORTALEZA
FRONT END	
BACK END	
INFRAESTRUCTURA	
TESTING/QA	
BASE DE DATOS	

## Patricia Del Castillo

Vivo en Córdoba ,trabaje de Acompañante de adulto mayor a la noche y en el dia empleada doméstica ,siempre me gusto la Tecnología,pero por mi familia no podía estudiar y en 2021 comencé en DH y al mes puede conseguir trabajo en Ayi Group , una empresa IT,haciendo una parte con backen y otra con base de datos y trabajo

Este primer año que curse en DH aprendí bastante ,y el Proyecto Integrador , estuve colaborando en la Parte de Frontend y Backend ,fue excelente la experiencia de este PI

TECNOLOGIA	FORTALEZA
FRONT END	
BACK END	
INFRAESTRUCTURA	
TESTING/QA	
BASE DE DATOS	



# Metodología de trabajo

---

La manera en que funciona la metodología Scrum es muy sencilla: una vez que se han creado los equipos se reparten el trabajo en una lista de pequeños entregables con un orden de prioridad; los tiempos de entrega se dividen en ciclos conocidos como Sprints, que por lo regular representan una semana en nuestro caso fueron de dos semanas.

Todos los integrantes colaboran en función de sus conocimientos individuales y el trabajo se optimiza a través de diferentes reuniones que se tienen diariamente, semanalmente al término de cada sprint.

El objetivo principal de la metodología Scrum es involucrarse con los clientes, el mercado y la tecnología a través de pequeñas acciones que ayuden a aumentar la productividad y la calidad de los productos y, sobre todo, lograr un mayor impacto comercial.

Como parte positiva de esta metodología vemos que ayuda a comunicarse mejor entre los miembros del equipo ya que se tienen que hacer reuniones diarias en donde se ven los progresos y las dificultades.

También fomenta el trabajo en equipo ya que es el equipo el que cumple la tarea y no solo el individuo y si un individuo no cumple con su tarea perjudica al equipo.

Propicia un marco ágil que permite cambiar de rumbo o de enfoque cuando el proyecto así lo necesita sin necesidad de llegar al final del mismo.

Como parte negativa vemos que muchas de las tareas dependen de que otras tareas estén realizadas previamente. Pero estas tareas están diseñadas para realizarse todas en el mismo Sprint dando menos tiempo a los miembros del equipo cuyas tareas dependen de la culminación de tareas de otros miembros del equipo.

También es difícil la división de tareas cuando muchas de ellas necesitan que otras tareas estén realizadas previamente. Encontramos dificultad en los primeros Spring en la división de tareas por esta causa Luego fueron corregidos en los Spring siguiente a analizar detenidamente cada tarea y tratar que un mismo miembro del equipo realice la tarea principal y las asociadas a un modulo del programa.

En nuestro trabajo el Product Owner fue Diego , el fue la persona encargada de determinar las prioridades del proyecto y representa a la empresa o usuarios. La Facilitadora o Scrum master en este caso fue Clara Irma Paredes que es nuestra tutora en Digital House.

El equipo de desarrollo está conformado por los integrantes del grupo-07.

Nos dividimos las tareas según preferencias y aptitudes.

Juaquin Marmol fue el encargado de toda la infraestructura desde el Spring 1 y también ayudó con la parte de estilos css en el Front.

Fernando Pajon fue el encargado de realizar casi todas las tareas de Testing y realizó también algunas tareas de Front.

Patricia del castillo realizó sobre todo las tareas de Front y también realizó algunas tareas de Back.

Katyana Prieto realizó sobre todo tareas de Front y también algunas de Back y de Testing.

Nicolas Gambino realizó sobre todo tareas de Back y también realizó algunas tareas de Front.

Karen Cespedes realizó sobre todo tareas de Back y también realizó algunas tareas de Front y el diseño del Figma que no estaba hecho para el modelo de autos.

Como acuerdos internos una vez que cada miembro del equipo terminaba con sus issues se ponía a servicio del resto para ayudarlo a completar sus issues o a disposición del equipo para los issues que debíamos arreglar o mejorar ya que todos entendimos que este proyecto era un trabajo en equipo y que el objetivo final era tener un buen producto.

También acordamos que si algún miembro del equipo trababa o encontraba dificultades en alguno de sus issues tenía que advertir al grupo para que todos veamos cómo solucionarlo.

# Bitácora del proyecto

---

Cada Sprint se vivieron de manera diferente el primero fue en el cual conocimos al equipo y fue un sprint sin demasiados issues el cual nos permitió conocernos un poco más y saber los puntos fuertes de cada integrante.

El segundo Sprint fue uno de los más complicados porque había muchos issues y repartimos las tareas sin tener en cuenta que unas necesitaban de que estén terminadas otras , lo cual fue bastante desorganizado y estresante en el aspecto de front y back.

Para el tercer Spring logramos repartir correctamente las tareas asignando dos integrantes solamente para back y dos integrantes para front. Tuvimos muchas dificultades con los TL ya que nos cancelaron las reuniones y tuvimos que recurrir a compañeros de la camada para resolver las dudas técnicas.

En el cuarto Spring al tener pocos issues y ya saber cómo organizarnos no tuvimos muchas dificultades.

Como fortalezas del equipo creo que todos trabajamos y nos ayudamos mutuamente. Pudimos apoyarnos si alguno por motivos personales necesitaba cambiar algún horario o no podía asistir a alguna reunión. Todos colaboramos con los issues de los otros y buscamos la manera de solucionar las dificultades en equipo.

Creo que una de las dificultades que tuvimos fue en una presentación que no llegamos a subir todo el contenido antes y lo hicimos rato antes de la presentación o mientras presentaban otros equipos. Si bien en esa oportunidad lo pudimos hacer y todos ayudaron para que saliera bien creo que aprendimos que no se sube nada el día de la presentación y que es preferible mostrar lo que hay que hacer cambios de último momento.

# Tecnologías utilizadas

Las tecnologías utilizadas en el proyecto fueron las siguiente:

- Gestión del proyecto: para gestionar el proyecto usamos un excel con las issues , descripción , a qué área corresponde, dificultad de la tarea y a quien le correspondía utilizarla.

	A	B	C	D	E	F	G	H
48	sprint 3	Crear la tabla "roles" en la base de datos	Issue 47	sprint 3	S	✓	Karen	Back
49		Crear la tabla de "usuarios" en la base de datos	Issue 48	sprint 3	S	✓	Karen	Back
50		Agregar registro de usuarios a la API	Issue 49	sprint 3	M	✓	Karen	Back
51		Conectar formulario de registro de usuarios con API	Issue 50	sprint 3	M	✓	Patri	Front
52		Conectar formulario de login con API	Issue 51	sprint 3	M	✓	Patri	Front
53		Agregar autenticación con Spring Security a endpoint de usuarios	Issue 52	sprint 3	XL	✓	Nico	Back
54		Conectar botón "Cerrar Sesión" con API	Issue 53	sprint 3	L	✓	Patri	Front
55		Implementar búsqueda por fecha desde la página principal	Issue 54	sprint 3	L	✓	Patri	Front
56		Implementar filtro por ciudad e intervalo de fechas en API	Issue 55	sprint 3	XL	✓	Nico	Back
57		Dar funcionalidad al botón de reserva en página de detalle de producto	Issue 56	sprint 3	M	✓	Karen	Front
58		rear tabla de "reservas" en la base de datos	Issue 57	sprint 3	S	✓	Nico	Back
59		Mapear las tablas "rol", "usuarios" y "reservas" con clases de nuestro modelo.	Issue 58	sprint 3	S	✓	Nico	Back
60		Crear endpoints de reservas en API con JpaRepository	Issue 59	sprint 3	M	✓	Nico	Back
61		Agregar seguridad con token al endpoint de creación de reservas	Issue 60	sprint 3	XL	✓	Nico	Back
62		Implementar el template pantalla de reserva	Issue 61	sprint 3	M	✓	Katy	Front
63		Implementar el formulario de reserva	Issue 62	sprint 3	M	✓	Katy	Front
64		Implementar submit del formulario de reservas	Issue 63	sprint 3	M	✓	Katy	Front
65		Implementar un calendario interactivo en la página de reservas	Issue 64	sprint 3	L	✓	Katy	Front
66		Implementar el template de reserva exitosa	Issue 65	sprint 3	M	✓	Katy	Front
67		Deploy en AWS	Issue 66	sprint 3	M	✓	Joaco	Infra
68		Implementar tests manuales	Issue 67	sprint 3	L	✓	Fernando	Testing

- Desarrollo del proyecto:
  - IDEs
  - Sistemas de control de versiones: usamos GITLAB
  - Simulación / Virtualización
- Qué programas utilizaron para el desarrollo del sitio
  - Front end : Visual Studio Code
  - Back end : **IntelliJ IDEA**
  - Base de datos:MYSQL
  - Infraestructura: Amazon ,
  - Testing :Postman, Selenium
- Qué programas utilizaron para el desarrollo del sitio

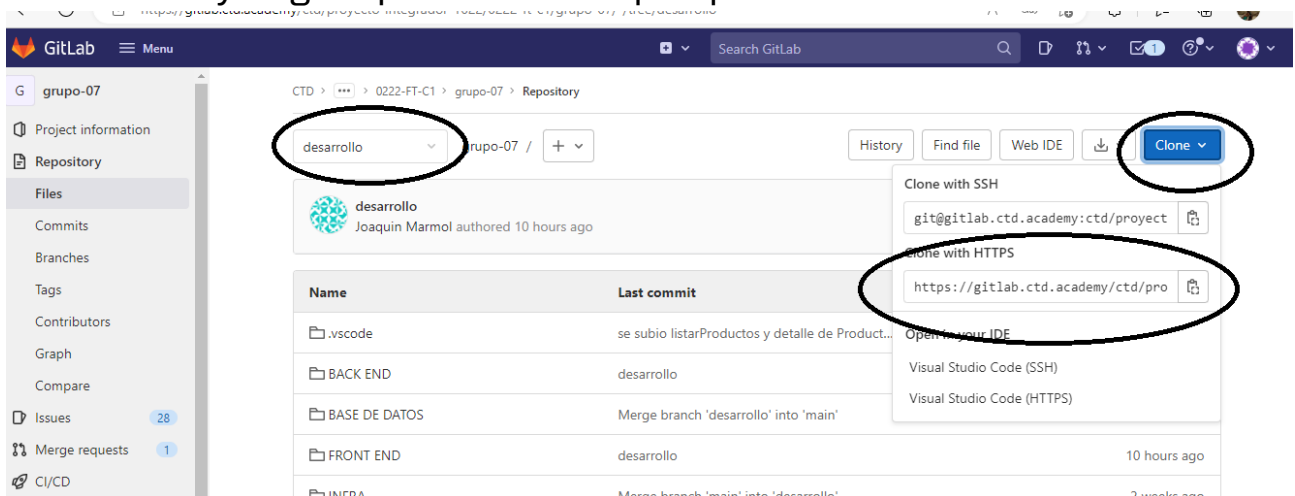
- Front end : React
- Back end : **Spring Boot ,JWT**

# Documentación técnica del proyecto

Si quieres ser un nuevo integrante de nuestro proyecto como primer medida deberás descargarte el proyecto en tu ordenador . crea una carpeta con el nombre del proyecto en el lugar del ordenador donde sea de tu preferencia. Luego abre el git bash here desde esa carpeta y clona nuestro proyecto desde la rama de desarrollo

\$ git clone "dirección del proyecto"

Para ver la dirección del proyecto entra en gitlab.Asegurate que estes en la rama de desarrollo y luego copia la dirección que aparece en el botón de clone.



Luego de tener el repositorio en tu ordenador podrás trabajar en él .

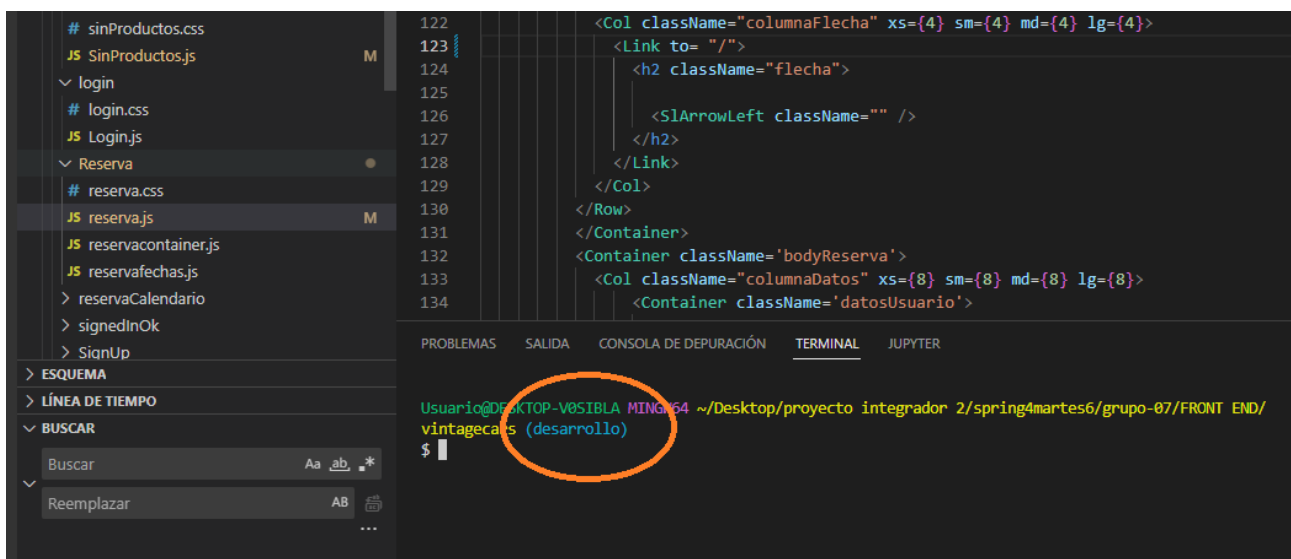
Cada día, siempre que iniciemos nuestro trabajo de forma local, se recomienda ejecutar:

```
$ git pull
```

Cada que agreguemos un nuevo archivo a nuestro proyecto, y sobre el cuál deseamos que se lleve un control de versiones, lo añadimos a git mediante:

```
$ git add mi-nuevo-archivo o $ git add .
```

Para guardar estos cambios hazlo en la terminal del visual studio code o directamente en la carpeta en el git bash. Una vez más asegúrate de estar en la rama desarrollo.



Este nuevo archivo es una mejora que agrega, aumenta, o corrige alguna funcionalidad, la descripción de dicha mejora debe añadirse a git

```
$ git commit -m "esta mejora permite que el usuario pueda hacer algo que antes
```

```
no podía"
```

Se descriptivo en cada commit o describir lo mejor posible la acción que se realiza en cada commit teniendo en cuenta esta frase: *“Si aplico este commit, entonces este commit...”*, no usar puntos en la descripción.

Cada que se modifique nuestro archivo, y estemos comprometidos con dicha modificación, repetimos los dos pasos anteriores, `git add`, `git commit -m``

Siempre es mejor tener muchos commits, que describan todo de forma detallada, tratando de evitar un gran commit que describa muchos cambios.

También es recomendable empujar los cambios frecuentemente, al menos una vez al día, con:

```
$ git push
```

Luego haz un `git pull` para traer todos los cambios que tus compañeros hayan realizado.

Si tu código actual entra en conflicto con el de algún compañero tendrás que hacer un merge antes de hacerlo ponte en contacto con tus compañeros para no borrar el trabajo de alguien más y decidir entre todos cuál es la opción correcta que deberá guardarse.

Si no hay ningún conflicto o se solucionaron los conflictos con el merge puedes subir tu código a la rama de desarrollo con el `git push origin desarrollo`.

Una vez que todos estén de acuerdo con el trabajo que está realizado en desarrollo este podrá subirse a la rama main.

📌 Al finalizar el desarrollo, cuando la funcionalidad está lista. Se solicita un **merge/pull request** de la **branch-con-nueva-funcionalidad** hacia **main** Merge request.

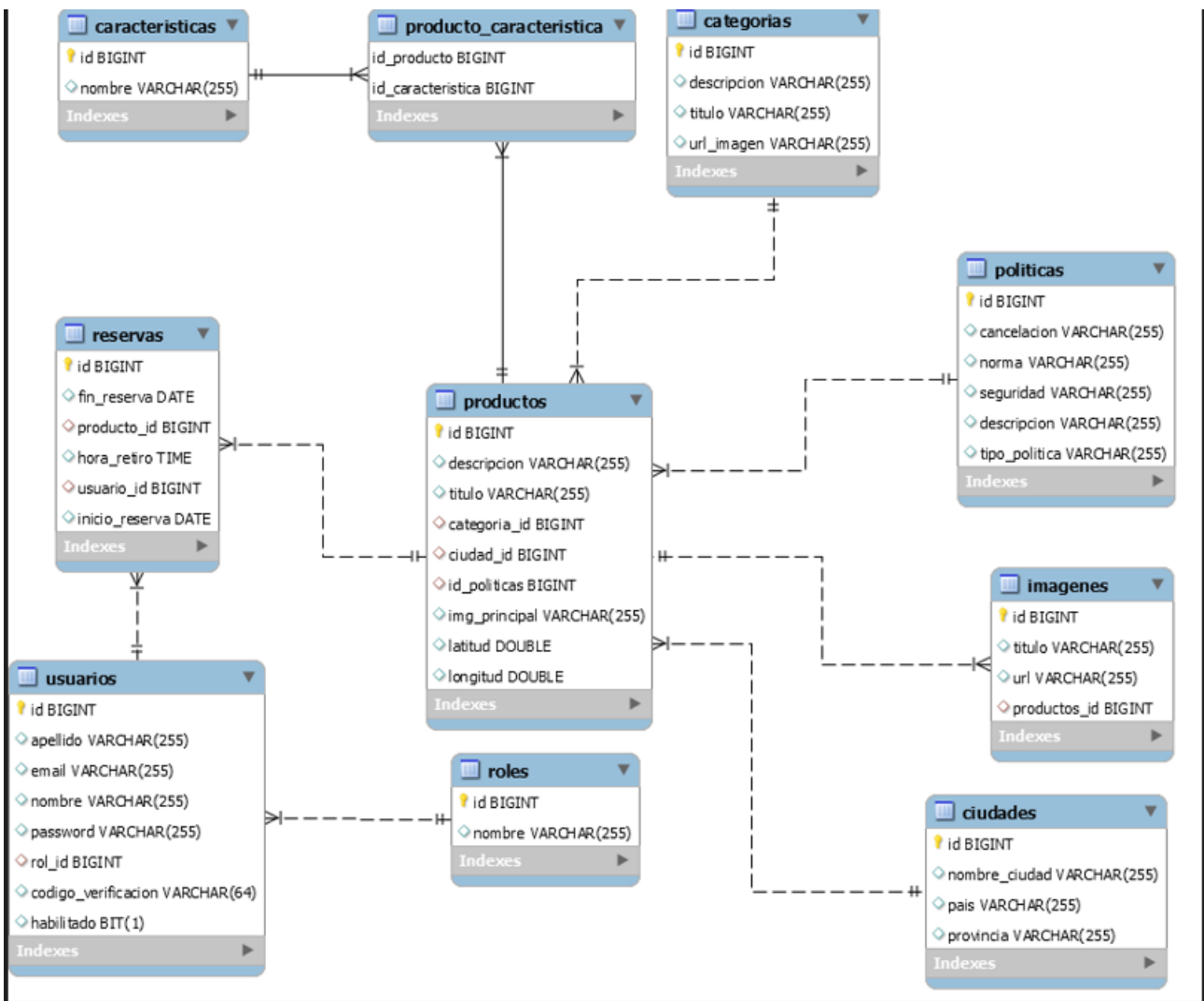
📌 Otro miembro del equipo debe hacer el **code review**, y, después de **solicitar las mejoras pertinentes al código**, puede **aceptar el merge request**. Después de aceptar el merge request se puede destruir la branch en la que se desarrolló la funcionalidad. Es usual que en este punto se encuentre **conflictos en el código**, el desarrollador y el revisor deben **consensuar** cómo resolver esos conflictos.

- Back End

En lo que es el Back End se utilizó el lenguaje de programación java, este proyecto se creo con spring boot utilizando el spring boot email y security, el primero para verificar la cuenta una vez creada y el segundo para tener seguridad en nuestra pagina por ejemplo encriptando las contraseñas y adquiriendo un jwt si los datos son correctos para poder navegar en la página. Está conectada a una base MySql creada por nosotros donde se encuentran todos las tablas con sus respectivos datos. Otra cosa que usamos fue el log4j para poder ir comentando lo que realiza nuestro programa en cada ejecución

- Base de datos

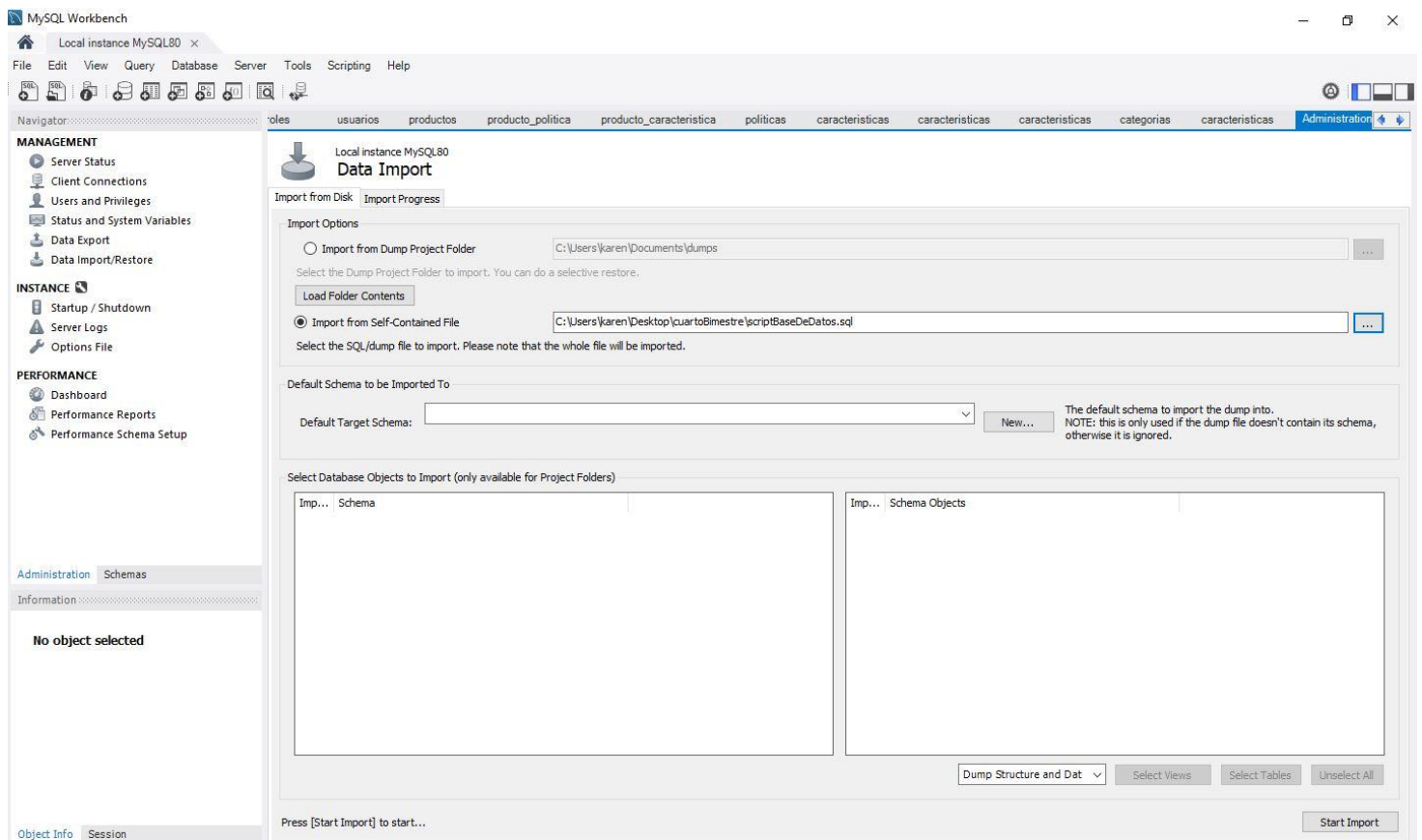
## Estructura y diagramas





- Scripts / instrucciones para la creación de la base de datos

Para la creación de la base de datos se necesita ir a la parte de administrador e importar en Mysql el archivo .sql que se encuentra en la carpeta base de datos.



Una vez importados se va a la solapa de schemas y actualizando se encontrará la base de datos creada.

- Scripts / set de datos dummy

Para conectarte a nuestra base de datos alojada en la nube se necesita que en MySql Connections crees una nueva conexion con las contraseñas asignadas por el grupo.

En nuestro caso necesitarás de un link para el hostname, un usuario y un password.

Con estos datos podrás conectarte en la base de datos.

- Componentes y diagramas de la aplicación

- Principales partes de la aplicación y cómo se conectan

Nuestra base de datos se conecta con el backend a través de un archivo llamado properties en el cual configuramos conexión a la base de datos y una vinculación con el frontend para el envío de mails.

- APIs disponibles y su documentación (pueden linkear a un sitio externo)

Nuestra api disponible con sus respectivos endpoints en la documentación

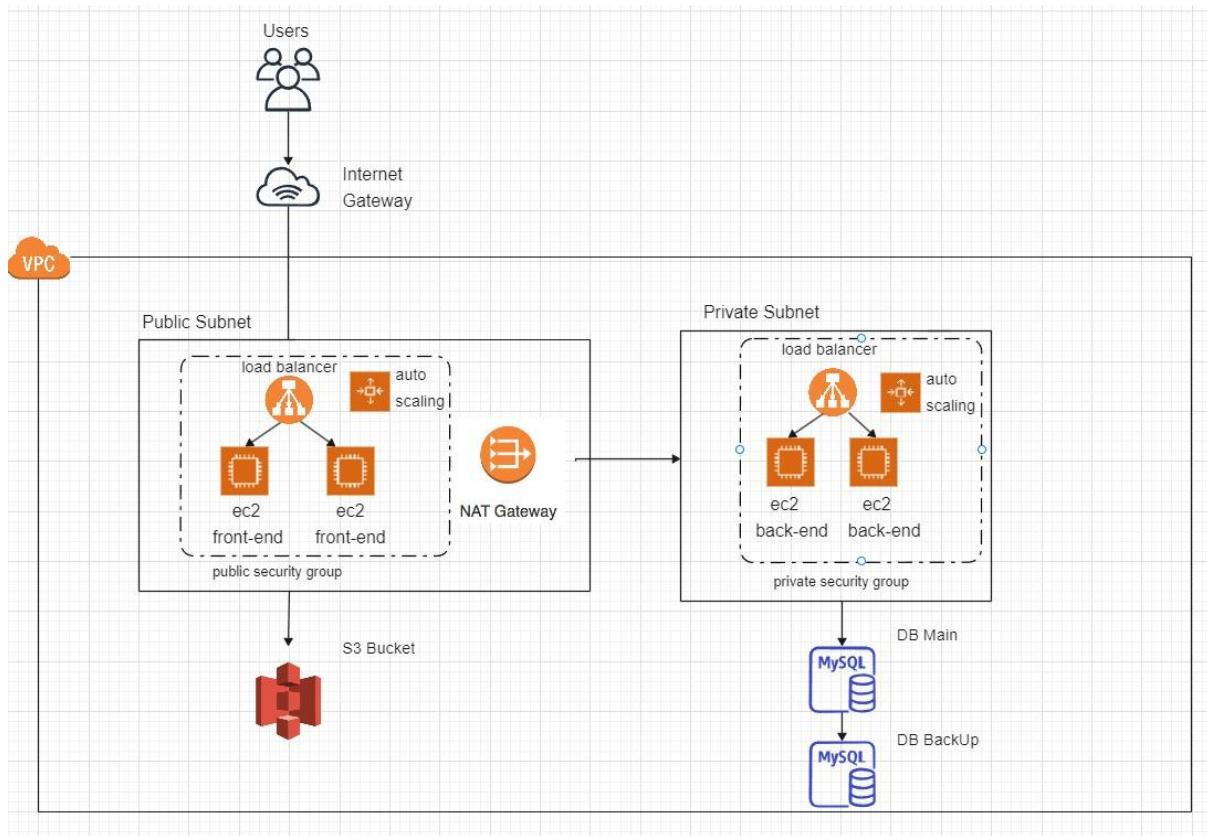
<http://3.14.134.132:8080/vintage/swagger-ui/index.html>

En la documentación se encuentran los endpoint para cada petición con los respectivos datos necesarios.

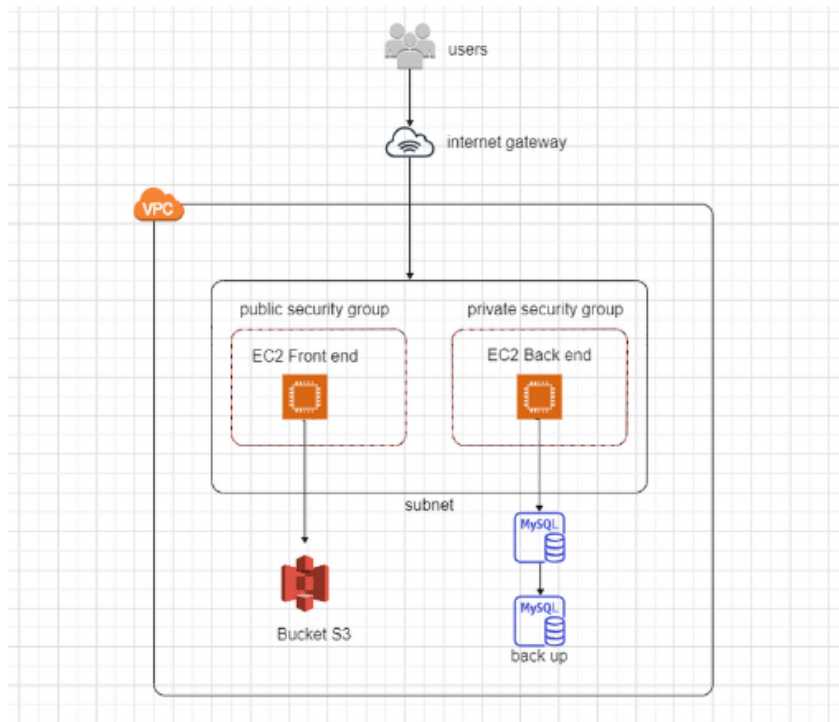
- Infraestructura

- Estructura y diagramas

En el primer sprint creamos un diagrama para la infraestructura necesaria para hacer funcionar el proyecto en AWS.



En el sprint 2 llevamos a cabo la creación de una VPC global, en la cual creamos las instancias (EC2), una para el Back y una para el Front. Cada instancia fue creada con sus respectivos niveles de seguridad, en el Back implementamos una subred privada con un grupo de seguridad privado y en el Front creamos una subred pública con un grupo de seguridad público.



Por último creamos un bucket s3 para almacenar nuestras imágenes para luego ser utilizadas en nuestra base de datos.

Luego en el sprint 3 se nos pidió que creamos un pipeline CI/CD para automatizar el despliegue del Back y del Front a la nube que después de varios intentos lo pudimos lograr.

Y por último, en el sprint 4 creamos nuestra propia URL con un certificado para poder acceder a nuestra página web de forma segura. (<https://vintagecar.sytes.net/>)

- Testing y calidad
  - Incluir el informe requerido para el cuarto sprint y cualquier otra información referente a cómo buscaron asegurar la calidad del software producido

#### TAREAS DE TESTING COMUNES A TODOS LOS SPRINTS:

El informe final se encuentra en la carpeta de gitlab Testing , Sprint 4 , informe final :

<https://gitlab.ctd.academy/ctd/proyecto-integrador-1022/0222-ft-c1/grupo-07/-/blob/main/TESTING/Sprint%204/Informe%20final%20de%20Testing%20-%20VintageCar.pdf>

**Informe final de Testing.pdf**



**Certified Tech  
Developer**  
The Ultimate Degree

**DigitalHouse** >  
Coding School

# -VintageCar-

## Reporte Final de Testing

### Grupo 7

#### Integrantes:

- Katyana Prieto
- Nicolás Gambino
- Joaquín Mármol
- Patricia Del Castillo
- Karen Céspedes Rivas
- Fernando Pajón

## **-Introducción-**

El presente documento es el informe final de las actividades de testing llevadas a cabo durante el desarrollo de *Vintage Car*, el proyecto integrador final del primer año de la carrera Certified Tech Developer (CTD) de Digital House, llevado a cabo por el grupo siete.

El propósito final de este instrumento es mostrar las actividades realizadas desde el primer sprint y poner de manifiesto que los criterios de aceptación (exit criteria) fueron cumplidos y, por lo tanto, esta fase puede cerrarse. A su vez, el objetivo ulterior es también servir como herramienta para determinar si el sistema cumple con las expectativas requeridas.

### Resumen de las actividades de prueba

¿Qué es *Vintage Car*? Se trata de una plataforma que permite alquilar coches de estilo.

Para el backend y la base de datos se realizaron pruebas unitarias de integración a través de postman. Para el frontend, se ejecutaron tests manuales, exploratorios y se utilizó Selenium para validar el funcionamiento de algunos componentes.

Nro	ISSUE	SPRINT	ESTADO
12	Planificación y ejecución de los test.	1	Cerrado
13	Testear la API.	1	Cerrado
46	Testeo automatizado.	2	Cerrado
48	Testeo con Postman.	2	Cerrado
49	Testeo exploratorio, manual, estático.	2	Cerrado
75	Implementación tests manuales	3	Cerrado
76	Implementar tests automatizados	3	Cerrado
87	Implementar testing automatizado	4	Cerrado
88	Implementar testing manual	4	Cerrado



## **-Alcance-**

### Dentro del alcance

Con testeo manual se probó la correcta implementación y gestión de ciudades, características del producto, el formulario de reserva, los inputs de usuario y contraseña, la visualización de las distintas cards, el footer y las tablas de la base de datos (aunque también se automatizó parte de estas pruebas con postman).

Con testeo automatizado se probó y verificó el CRUD de la API, el Security y la integración del sistema.

También se probaron las funciones referidas a la creación de usuario, al inicio de sesión erróneo e incorrecto, el calendario, la existencia del template reserva, la correcta reserva, el formulario de producto, el bloque buscador, el enrutador al home, eventos de login, responsive calendario, la visualización de productos y descripciones, entre otros.

### Fuera de alcance

Quedaron fuera de alcance los test relativos a la confirmación de cuenta de usuario, a la latitud y longitud del mapa, y lo relativo a pruebas de seguridad y estrés.

### Tipos de pruebas ejecutadas

	SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 4
Prueba estática	✓	✓	✓	✓
Prueba exploratoria	✓	✓	✓	✓
Prueba de humo	✓	✓	✓	✓
Prueba de regresión	X	✓	✓	✓
Prueba de componente	✓	✓	✓	✓

### **-Enfoque de la prueba-**

Inicialmente se leía detenidamente el backlog de cada sprint y nos las dudas que pudiéramos tener las despejábamos con el product owner durante la planning.

A continuación, en una planilla de hoja de cálculo, disponíamos las issues y la dividíamos equitativamente, pero sabiendo que en cualquier momento podíamos rotar en nuestras tareas si era necesario para asegurar que el proyecto avance.

Desde el área de testing, realizamos test negativos y positivos a lo largo de todo el desarrollo. Creamos una rama para subir al repositorio todo lo relativo al testing. Durante el transcurso de primer sprint, nos dedicamos al testeo puramente manual porque nos encontrábamos todavía en el momento de diseño de la página, por lo que no había regresión para testear o automatizar.

Durante el segundo sprint se dió la posibilidad de usar jest. Aunque nuestro equipo optó al principio por implementarlo en el desarrollo de nuestra página, y logramos usarlo para testear componentes unitarios como algunas card, filtros y demás, luego lo descartamos por problemas de configuración, puesto que jest entraba en conflicto con algunas dependencias (packaje.json) y herramientas relacionadas a react.

En este sprint, todo se probaba en localhost, llevando a cabo testeo manual y automático de las funcionalidades más relevantes, quienes eran examinadas positiva y negativamente para asegurar la mayor calidad y descartar errores “ocultos” o difíciles de encontrar si solo hacíamos happy path.

El tercer sprint fue probablemente el más desafiante y el que más requirió de nuestro esfuerzo. El uso de postman

para testar la API luego de que el back añadiera seguridad fue fundamental para asegurar la calidad exigida.

El front end siguió testeándose fundamentalmente de manera manual y con testeo de regresión a través de Selenium Web Driver. En estos sprints, iba incrementándose obviamente la cantidad de test de regresión y con un incremento en la cantidad de casos de prueba enfocados en las funcionalidad primordiales logramos abarcar en un alto porcentaje las issues requeridas, quedando solo algunos carry overs para el siguiente sprint.

El último sprint encontró a los responsables de testing, front end y back end con una comunicación más fluida y con mayor gimnasia a la hora de desarrollar y asegurar calidad. Así, cada vez que una build era entregada se testeaba extensamente y se realizan sesiones de testeo exploratorio donde se dejaba constancia de los resultados o los posibles defectos que pudieran haberse encontrado.

Issues ☆ 📁 📧

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

↶ ↷ 🖨 📌 100% € % ,0 .00 123 ▾ Predetermi... 9 ▾ B I A 📄 🗒 📑 ⌵ ⌶ ⌷ ⌸ ⌹ ⌺ ⌻ ⌼ ⌽ ⌾ ⌿

## -Exit Criteria-

Fueron establecidos como criterios de salida, los siguientes:

Se definieron los siguientes criterios de aceptación para finalizar las pruebas:

- No debe tener defectos en estado abierto de severidad crítica y/o bloqueante.
- No se debe tener defectos en estado abierto de prioridad crítica.
- La ejecución de las pruebas de regresión deben tener mínimo un 80% de éxito.
- Los tests de postman de issues esenciales requieren 100% de éxito.

### -Resumen de resultados-

	Test Manuales	Test automáticos	Test de integración con Postman
Login usuario	3	2	3
Creación de cuenta	4	2	3
Seleccionar Producto	2	0	6
Reserva	2	3	9
Formulario Crear Producto	3	2	5
Filtros productos	2	2	4
Responsive	6	0	0
Calendario	2	1	0
Botones	1	0	0
Tablas Bases de Datos	8		12
Seguridad API	4	1	6
Form reserva	5	4	3

	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ ...	Persist All	Reset All
<input checked="" type="checkbox"/>	BASE_URL	default ▾	http://3.14.246.73:8080/vint...	http://3.14.246.73:8080/vintage		
<input checked="" type="checkbox"/>	AUTH_URL	default ▾	{{(BASE_URL)}}/authenticate	{{(BASE_URL)}}/authenticate		
<input checked="" type="checkbox"/>	JWT	default ▾		eyJhbGciOiJIUzI1NiJ9.eyJ2ZXIpZmllY...		
	Add a new variable					

Proyecto integrador - grupo 7 / Authentication / Post authenticatce

POST ▾ {{AUTH\_URL}}

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

```
1 pm.environment.set("JWT", pm.response.json().jwt)
```

Test scripts are written in JavaScript, and are run after the response is received. Learn more about [tests scripts](#)

Snippets

[Get an environment variable](#)

[Get a global variable](#)

Body Cookies Headers (14) Test Results

200 OK 488 ms 737 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾ ↺

```
1 {
2   "jwt": "eyJhbGciOiJIUzI1NiJ9.eyJ2ZXIpZmllYWRvIjpbOcnVlLCJzdWIiOiJmZXJuYW5kb3Bham9uMDAZZ21haWwuyJ29tIiwibmFtZSI6IktZlcmShbmRvIiwibGZdF09uYWllIjoiaUGFqb241LClCjpZCI6MzIsImV4cCI6MTY3MDQ1OTgxNSwiaWF0IjoxNjcwNDU2MjE1LCl3cyw2aW0nsiaWQiOiJESIm5vbWJyZSIlbjPTEVFjQRNSU4ifX0.yVSpIATcgU0JMIB_msZCYVjwGRtz_U3d4u3D7GNBYnw"
```



Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	2s 479ms	2	304 ms

All Tests

Passed (2)

Failed (0)

Skipped (0)

[View Summary](#)

teration 1

POST

Post authenticathe

{{AUTH\_URL}}

/ Authentication / Post authenticathe

200 OK

327 ms

737 B

Pass

Status code is 200

POST

Post authenticathe (USUARIO SIN PERMISOS)

{{AUTH\_URL}}

/ Authentication / Post aut...

403 Forbidden

281 ms

551 B

Pass

Status code is 403

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	1s 757ms	4	241 ms

All Tests

Passed (4)

Failed (0)

Skipped (0)

[View Summary](#)

teration 1

GET

Get características

{{BASE\_URL}}/caracteristica

/ características / Get características

200 OK

363 ms

1.116 KB

Pass

Status code is 200

POST

Post características

{{BASE\_URL}}/caracteristica

/ características / Post características

200 OK

191 ms

464 B

Pass

Status code is 200

PUT

Put características

{{BASE\_URL}}/caracteristica

/ características / Put características

200 OK

211 ms

477 B

Pass

Status code is 200

DELETE

Del características

{{BASE\_URL}}/caracteristica/37

/ características / Del características

200 OK

200 ms

473 B

Pass

Status code is 200

[Cookies](#)

[Capture requests](#)

[Runner](#)

[Trash](#)

[🔍](#)

[🗑️](#)

D



Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	3s 103ms	5	247 ms

All Tests Passed (5) Failed (0) Skipped (0)

[View Summary](#)

Iteration 1

<b>GET</b>	Get categorias	{{BASE_URL}}/categorias	/ categoria / Get categorias	200 OK	387 ms	1.154 KB
	Pass	Status code is 200				
<b>POST</b>	Post categorias	{{BASE_URL}}/categorias	/ categoria / Post categorias	200 OK	212 ms	665 B
	Pass	Status code is 200				
<b>PUT</b>	Put categorias	{{BASE_URL}}/categorias	/ categoria / Put categorias	200 OK	220 ms	472 B
	Pass	Status code is 200				
<b>DELETE</b>	Del categorias	{{BASE_URL}}/categorias/35	/ categoria / Del categorias	200 OK	223 ms	468 B
	Pass	Status code is 200				
<b>POST</b>	Post categorias (NO ADMIN)	{{BASE_URL}}/categorias	/ categoria / Post categorias (N...	403 Forbidden	192 ms	549 B
	Pass	Status code is 403				

[Cookies](#) [Capture requests](#) [Runner](#) [Trash](#) [Grid](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	1s 975ms	4	298 ms

All Tests Passed (3) Failed (1) Skipped (0)

[View Summary](#)

Iteration 1

<b>GET</b>	Get imagenes	{{BASE_URL}}/imagenes	/ imagenes / Get imagenes	200 OK	567 ms	34.349 KB
	Pass	Status code is 200				
<b>POST</b>	Post imagen	{{BASE_URL}}/imagen	/ imagenes / Post imagen	200 OK	213 ms	589 B
	Fail	Status code is 200   TypeError: Cannot read properties of undefined (reading 'to')				
<b>PUT</b>	Put imagen	{{BASE_URL}}/imagen	/ imagenes / Put imagen	200 OK	208 ms	470 B
	Pass	Status code is 200				
<b>DELETE</b>	Del imagen	{{BASE_URL}}/imagen/180	/ imagenes / Del imagen	200 OK	204 ms	466 B
	Pass	Status code is 200				

[Cookies](#) [Capture requests](#) [Runner](#) [Trash](#) [Grid](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	1s 849ms	4	290 ms

All Tests

Passed (4)Failed (0)Skipped (0)

[View Summary](#)

Iteration 1

GET

Get politica

{{BASE\_URL}}/politica

/ politica / Get politica

200 OK

546 ms

14.125 KB

Pass

Status code is 200

POST

Post politica

{{BASE\_URL}}/politica

/ politica / Post politica

200 OK

199 ms

1.049 KB

Pass

Status code is 200

PUT

Put politica

{{BASE\_URL}}/politica

/ politica / Put politica

200 OK

208 ms

471 B

Pass

Status code is 200

DELETE

Del politica

{{BASE\_URL}}/politica/46

/ politica / Del politica

200 OK

205 ms

467 B

Pass

Status code is 200

Cookies

Capture requests

Runner

Trash

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	4s 118ms	9	289 ms

All Tests

Passed (9)

Failed (0)

Skipped (0)

View Summary

Iteration 1

GET

Get productos

{{BASE\_URL}}/producto / producto / Get productos

200 OK

629 ms

35.288 KB

Pass

Status code is 200

Pass

Response time is less than 200ms

GET

Get producto por ID

{{BASE\_URL}}/producto/1 / producto / Get producto por ID

200 OK

203 ms

2.911 KB

Pass

Status code is 200

Pass

Response time is less than 200ms

POST

Post productos

{{BASE\_URL}}/producto / producto / Post productos

200 OK

286 ms

2.757 KB

Pass

Status code is 200

Pass

Response time is less than 200ms

PUT

Put producto

{{BASE\_URL}}/producto / producto / Put producto

200 OK

205 ms

471 B

Pass

Status code is 200

DELETE

Del producto

{{BASE\_URL}}/producto/43 / producto / Del producto

200 OK

217 ms

467 B

Pass

Status code is 200

POST

Post producto (NO ADMIN)

{{BASE\_URL}}/producto / producto / Post producto (NO AD...

403 Forbidden

191 ms

547 B

Pass

Status code is 403

🍪 Cookies

📄 Capture requests

🏃 Runner

🗑️ Trash

⌵

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	4s 299ms	6	308 ms

[All Tests](#) [Passed \(6\)](#) [Failed \(0\)](#) [Skipped \(0\)](#) [View Summary](#)

Iteration 1

GET	Get reservas	{{BASE_URL}}/reserva	/ reservas / Get reservas	200 OK	781 ms	76.288 KB
	Pass	Status code is 200				
GET	Get reserva por ID	{{BASE_URL}}/reserva/2	/ reservas / Get reserva por ID	200 OK	207 ms	3.371 KB
	Pass	Status code is 200				
POST	Post reservas	{{BASE_URL}}/reserva	/ reservas / Post reservas	200 OK	226 ms	1.132 KB
	Pass	Status code is 200				
PUT	Put reservas	{{BASE_URL}}/reserva	/ reservas / Put reservas	200 OK	240 ms	470 B
	Pass	Status code is 200				
DELETE	Delete Reserva	{{BASE_URL}}/reserva/52	/ reservas / Delete Reserva	200 OK	208 ms	466 B
	Pass	Status code is 200				
POST	Post reserva (NO ADMIN)	{{BASE_URL}}/reserva	/ reservas / Post reserva (NO ADMIN)	403 Forbidden	188 ms	546 B
	Pass	Status code is 403				

[Cookies](#) [Capture requests](#) [Runner](#) [Trash](#) [Grid](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Autenticación global	1	3s 56ms	3	587 ms

[All Tests](#) [Passed \(3\)](#) [Failed \(0\)](#) [Skipped \(0\)](#) [View Summary](#)

Iteration 1

GET	Get usuarios	{{BASE_URL}}/usuario	/ usuarios / Get usuarios	200 OK	208 ms	8.215 KB
	No tests in this request.					
GET	Get usuarios por ID	{{BASE_URL}}/usuario/32	/ usuarios / Get usuarios por ID	200 OK	201 ms	849 B
	Pass	Status code is 200				
POST	Post usuario	{{BASE_URL}}/usuario	/ usuarios / Post usuario	201 Created	1720 ms	935 B
	Pass	Status code is 200				
DELETE	Del usuario	{{BASE_URL}}/usuario/59	/ usuarios / Del usuario	200 OK	220 ms	464 B
	Pass	Status code is 200				

[Cookies](#) [Capture requests](#) [Runner](#) [Trash](#) [Grid](#) [Printer](#)

## Selenium:

http://localhost:3000			
	Command	Target	Value
1	✓ open	/	
2	✓ set window size	704x502	
3	✓ type	name=myBrowser	Buenos aires
4	✓ type	name=myBrowser	Rosario
5	✓ type	name=myBrowser	Cordoba
6	✓ click	css=.contenedorbuscador	
7	✓ type	name=myBrowser	Mendoza
8	✓ click	css=.FechaRetiro	
9	✓ click	css=.react-datepicker__day--025 > font > font	
10	✓ click	css=.calendario2	
11	✓ click	css=.react-datepicker__day--030	
12	✓ click	css=.react-datepicker__day--030 > font > font	
13	✓ click	css=.contenedorbotones	
14	✓ click	css=.boton > font > font	

Running 'bloque buscador'

- 1. open on / OK
- 2. setWindowSize on 704x502 OK
- 3. type on name=myBrowser with value Buenos aires OK
- 4. type on name=myBrowser with value Rosario OK
- 5. type on name=myBrowser with value Cordoba OK
- 6. click on css=.contenedorbuscador OK
- 7. type on name=myBrowser with value Mendoza OK
- 8. click on css=.FechaRetiro OK
- 9. click on css=.react-datepicker\_\_day--025 > font > font OK
- 10. click on css=.calendario2 OK
- 11. click on css=.react-datepicker\_\_day--030 OK
- 12. click on css=.react-datepicker\_\_day--030 > font > font OK
- 13. click on css=.contenedorbotones OK
- 14. click on css=.boton > font > font OK

'bloque buscador' completed successfully

http://localhost:3000/

	Command	Target
1	open	http://localhost:3000/home
2	set window size	1382x744
3	click	xpath=//nav[@id='navbar']/div/a/span[4]

Running 'home'

1. open on http://localhost:3000/home OK

'home' completed successfully

http://localhost:3000/			
	Command	Target	Value
1	✓ open	http://localhost:3000/	
2	✓ set window size	702x500	
3	✓ click	css=#offcanvasNavbarLabel-expand-sm .btn-close	
4	✓ type	id=formBasicNombre	patri@gmail.com
5	✓ type	id=formBasicPassword	hola123
6	✓ mouse down at	css=html	-20,333
7	✓ mouse move at	css=html	-20,333
8	✓ mouse up at	css=html	-20,333
9	✓ click	css=html	
10	✓ click	css=.container	

Log	Reference
<b>Running 'evento login usuario'</b>	
1. open on http://localhost:3000/ OK	
2. setWindowSize on 702x500 OK	
3. click on css=#offcanvasNavbarLabel-expand-sm .btn-close OK	
4. type on id=formBasicNombre with value patri@gmail.com OK	
5. type on id=formBasicPassword with value hola123 OK	
6. mouseDownAt on css=html with value -20,333 OK	
7. mouseMoveAt on css=html with value -20,333 OK	
8. mouseUpAt on css=html with value -20,333 OK	
9. click on css=html OK	
10. click on css=.container OK	
11. click on css=.formularioButton OK	
<b>'evento login usuario' completed successfully</b>	

http://3.144.28.205

	Command	Target
1.	✓ open	/
2.	✓ set window size	1296x696
3.	✓ click	css= cardContainer:nth-child(1) a font > font
4.	✓ run script	window.scrollTo(0,0)
5.	✓ run script	window.scrollTo(0,599.3333129882812)
6.	✓ set window size	720x1240
7.	✓ click	css= botonReservaCalendario

'ResponsiveMobile' completed successfully

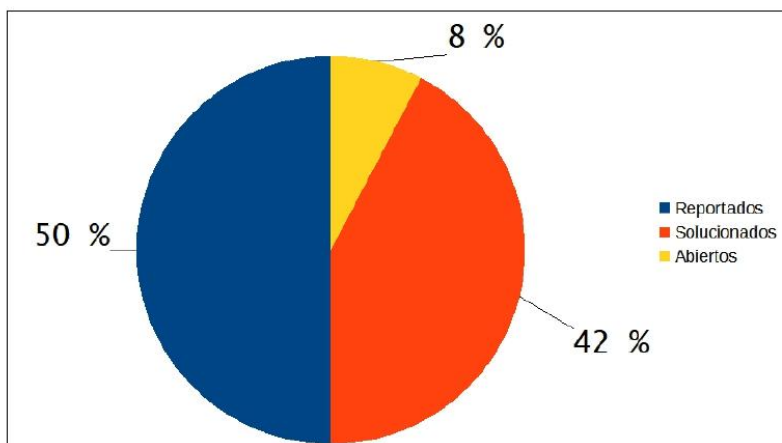
Running 'ResponsiveMobile'

- open on / OK
- setWindowSize on 1296x696 OK
- Trying to find css= cardContainer:nth-child(1) a font > font... OK
- runScript on window.scrollTo(0,0) OK
- runScript on window.scrollTo(0,599.3333129882812) OK
- setWindowSize on 720x1240 OK
- click on css= botonReservaCalendario OK

'ResponsiveMobile' completed successfully

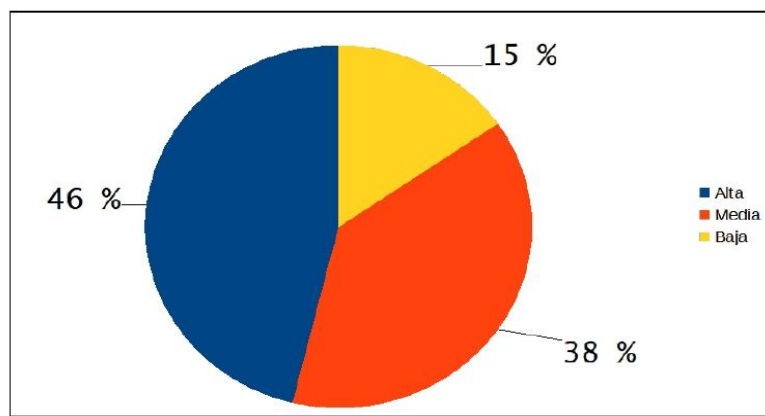
## -Reporte de Defectos-

### Defectos por estado:

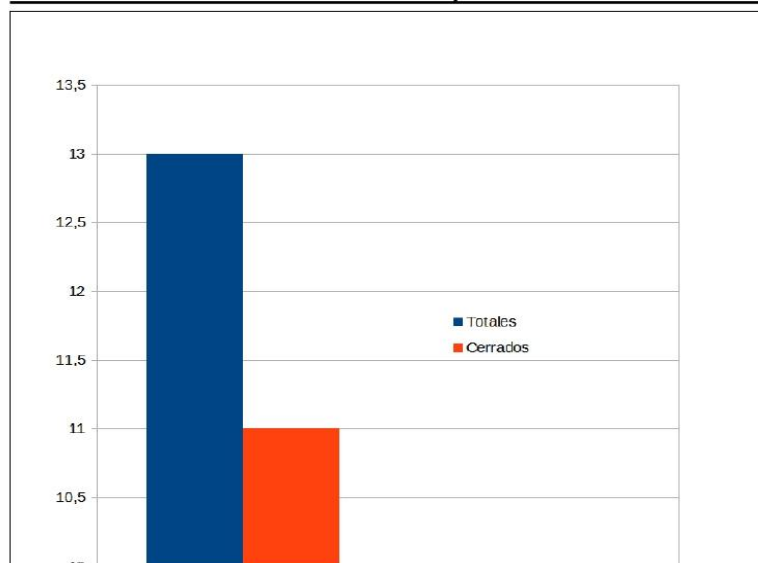




### Defectos por severidad



### Defectos totales creados / defectos cerrados.



### **-Lecciones Aprendidas. Conclusión-**

El recorrido por todos los sprints nos hizo comprender la gran importancia del testing durante el proceso de desarrollo a medida que se iban agregando e integrando más funcionalidades.

El testeo continuo fue de gran ayuda para encontrar, asignar y relevar mucho más rápidamente los defectos y poder actuar en consecuencia, permitiéndonos optimizar el tiempo, que es probablemente lo más importante, y la mejor estrategia que podemos seguir para lograr cumplir con las entregas de cada sprint.

La posibilidad de experimentar con herramientas que vimos en la cursada es una de las enseñanzas más gratificantes para quienes hicimos testing en este proyecto integrador, como así también haber afianzado los conceptos que aprendimos en clases. Explorar el producto desde diferentes ángulos, con distintas enfoques, ópticas y herramientas nos permitió entregar un producto con una calidad que, para nosotros que no teníamos experiencia en este rubro, nos satisface grandemente y no podemos

--  
dejar de agradecer a todo el equipo de Digital House por hacerlo posible.

Entendimos también lo primordial que es el testing en un proceso como este, convirtiéndose en algo totalmente fundamental para que el producto final tenga una calidad aceptable, con la menor cantidad de errores posibles, accesible, adaptable y con valor para el cliente.