



## PROYECTO PLATAFORMA RECOGIDA DE PAQUETES – “InstaYA”



## Documentación Proyecto

**Grupo NCR: 2252**

**Equipo No. 13**

**Link del GitHub:**

<https://github.com/Mininorelleno/frontend#readme>

**Integrantes Equipo 13:**

*Juan Pablo Medina Buitrago  
Jairo Antonio Aguirre Osorio  
Camilo Pérez Fonseca  
Aníbal Salom González  
Jorge Eliécer Vidal Rodríguez*

**08-11-2022**

**Tabla 1. Descripción de roles del equipo**

Rol	Integrante	Descripción	Tareas
SCRUM Master	Tutor	Persona encargada de gestionar y organizar el Scrum	Revisar el cumplimiento de la metodología Scrum. Revisar el correcto desarrollo de documentos a entregar
Product Owner	Tutor	Encargados de visionar y gestionar los objetivos a desarrollar en el proyecto, además, de mantener el Product Backlog y gestionar el Equipo	Revisar el cumplimiento de los puntos del proyecto. Revisar el correcto funcionamiento de la aplicación.
Desarrollador Front-End	Integrante 1 y 4	Personas encargadas de realizar el diseño de las vistas y experiencia de usuario	Realizar código HTML, CSS y JS (Front-End). Realizar las vistas del usuario.
Desarrollador Back-End	Integrante 1 y 2	Personas encargadas de realizar el desarrollo back-End incluida la conexión con la base de datos.	Conectar con Front-End. Generar las APIs necesarias. Conexión con BD para usar los diferentes métodos http.
Desarrollador Full-Stack	Integrante 1	Encargados de realizar la revisión de la conexión entre el Back-End y Front-End	Verificar la correcta conexión entre el Back-End y el Front-End Realizar las sugerencias necesarias para el desarrollo del Back-End y el Front-End
QA-QC Tester	Integrante 3 y 5	Encargados del diseño y ejecución de las pruebas. Mantienen el estándar de calidad del proyecto	Diseñar un plan de testing y definir los requerimientos paso a paso de las pruebas técnicas, funcionales o no funcionales. Despliegue y entrega del proyecto

# ARQUITECTURA Y DIAGRAMAS DE DESPLIEGUE

## DIAGRAMAS DE DESPLIEGUE AMBIENTE DE DESARROLLO

### 1. DIAGRAMA #1 DESPLIEGUE EN PC DESARROLLO

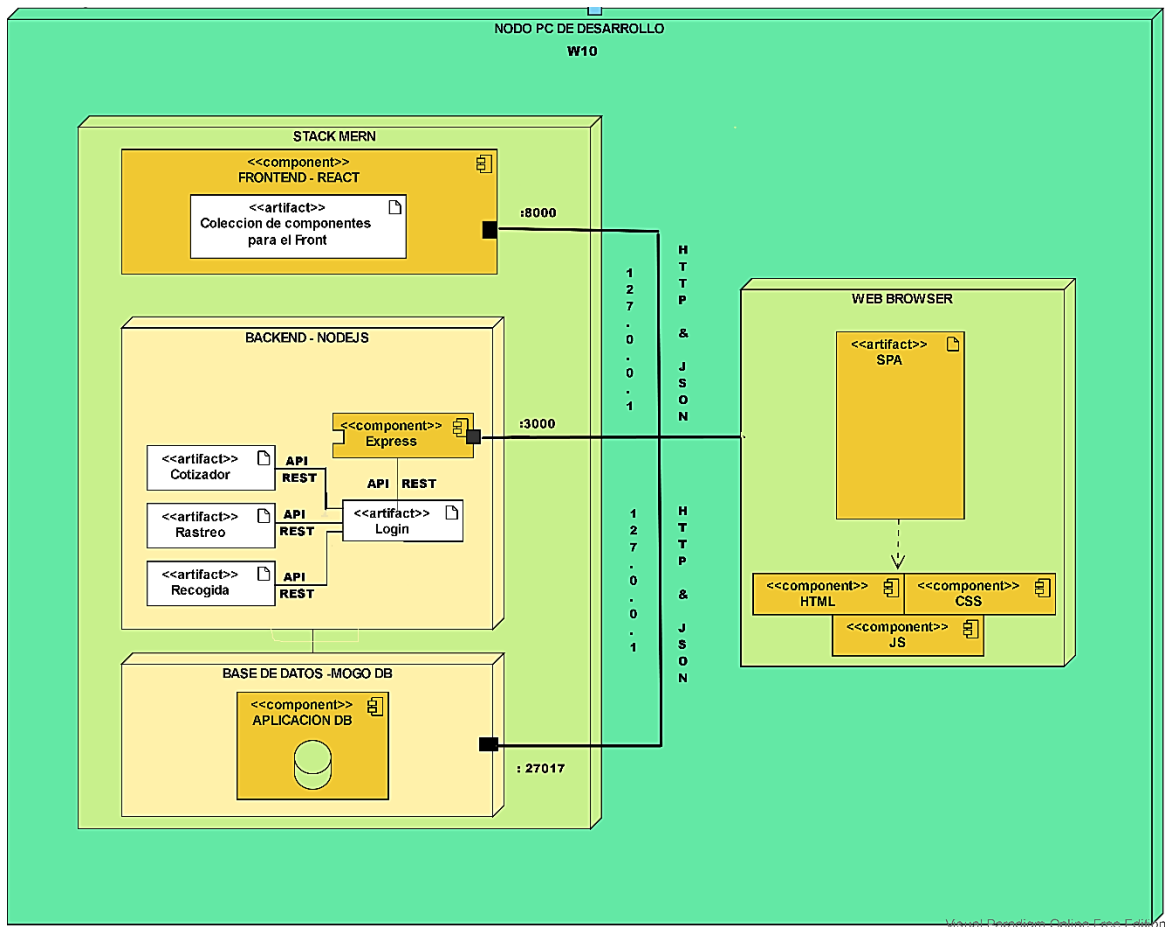


Imagen 1. Diagrama de Despliegue del Sistema de Reserva para el Proyecto InstaYA.

Este diagrama de despliegue, Imagen1, muestra la arquitectura sobre la cual es implementado el ambiente de desarrollo en el PC de cada uno de los desarrolladores. Puede apreciarse que el Core base de la aplicación corresponde a un Stack MERN, una imagen de este CORE será contenerizado para el despliegue final en producción de la aplicación. Por ello vale la pena detenernos un poco para analizar su estructura e interacciones.

El CORE está estructurado con los siguientes componentes:

- **MONGO DB:** sobre el cual corre la instancia de base de datos requerida por nuestra aplicación.

- **NODE.JS con EXPRESS:** constituyen el Back-End de la aplicación mediante microservicios, que ejecutándose a su interior implementan cada una de las funcionalidades de la aplicación, a la vez que enruta el tráfico http hacia ellas con base en la URL al cual va direccionado el requerimiento http. La comunicación con cada uno de los microservicios es posible a las API REST que implementan para poder ser utilizadas.
- **REACT:** es el framework sobre el cual se construye el Front-End de la aplicación mediante la modularización de la interface gráfica haciendo uso de componentes invocables y reutilizables; este componente se encarga de enviar al navegador la información y comandos necesarios para que este renderice con base en HTML, CSS y JavaScript, de forma local la SPA que representa la cara al usuario de la aplicación.

Todos estos componentes residen en el PC de cada desarrollador y se comunican bidireccionalmente mediante tramas TCP/IP direccionadas al localhost, que transportan requerimientos http y datos en formato JSON. Como todos comparten como destino el local host, cada componente se identifica mediante un puerto TCP/IP diferente, 3000 para Express – Node.js, 8000 para React y 27017 para el manejador de la base de datos. El navegador es el medio que materializa el acceso a la aplicación y permite visualizar el alcance de la implementación y probar funcionalidades.

## 2. DIAGRAMA #02 DESPLIEGUE EQUIPO DE DESARROLLO

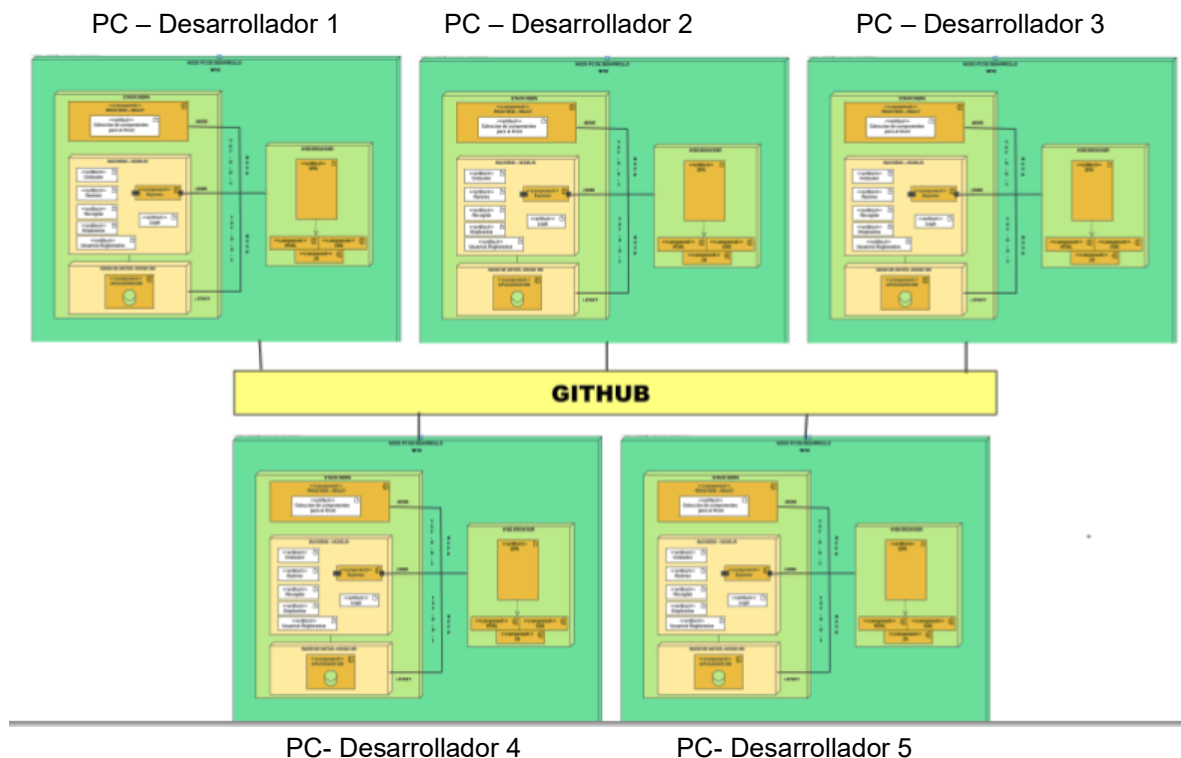


Imagen2. Diagrama de Despliegue del Equipo de Desarrollo.

Cada miembro del equipo de desarrollo trabaja en un PC que **REPLICA** la configuración detallada en el diagrama #01, GITHUB se utiliza como repositorio de software sobre el cual

cada desarrollador tiene asignada una rama a la que sube los incrementos funcionales en la medida que los va desarrollando, para finalmente consolidar las nuevas versiones de la aplicación mediante la fusión de estas con la rama master del proyecto, previo cumplimiento de los procedimientos establecidos.

### 3. DIAGRAMA #3 DESPLIEGUE EN AMBIENTE DE PRODUCCION

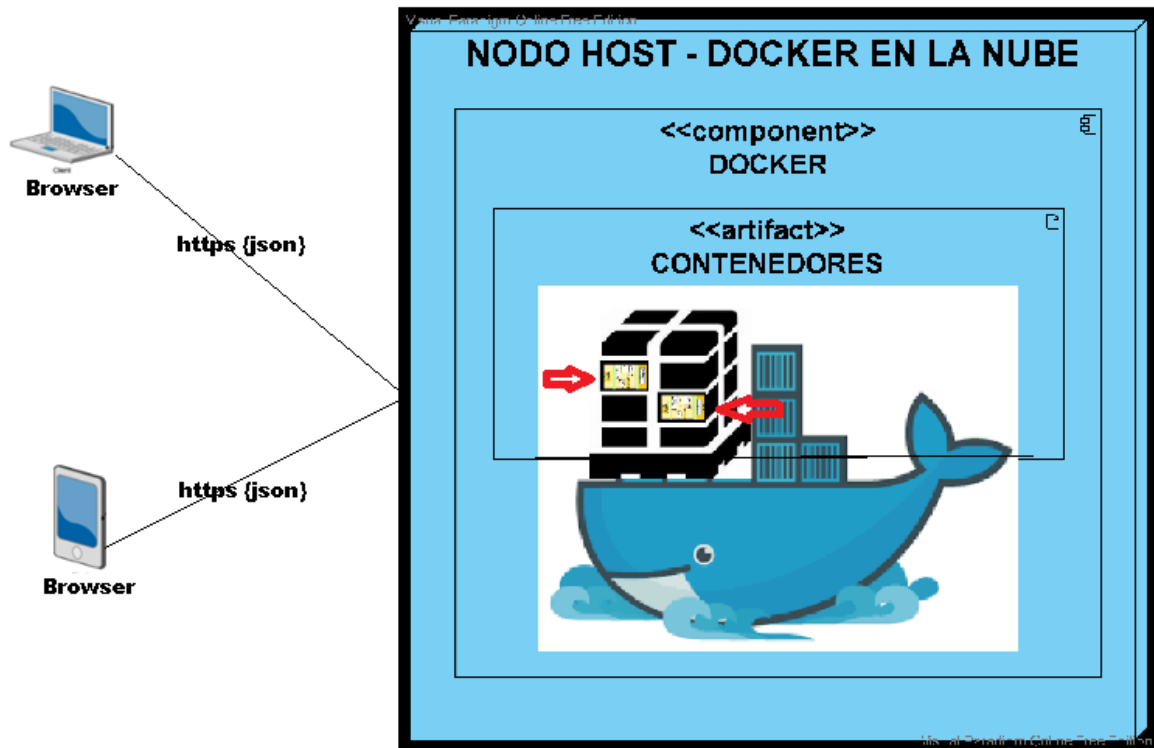
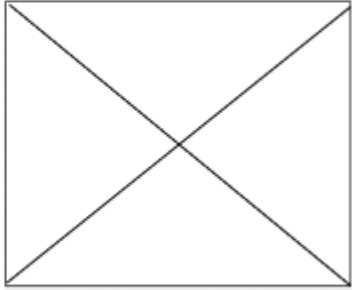


Imagen 3. Diagrama de Despliegue – Ambiente de Producción

Para el despliegue final de la aplicación a través del cual entra en producción, se conteneriza la versión final de desarrollo. Un contenedor aloja una imagen de la aplicación que incluye todos los requerimientos y dependencias necesarios para que funcione de forma autónoma. Un contenedor es una forma muy práctica de desplegar o trasladar una aplicación pues el contenedor a través de la imagen que aloja es totalmente autosuficiente, no requiriéndose instalar nada en el destino, ya que solo es necesario poner el contenedor dentro del ambiente docker/kubernetes de la nube para que este se ejecute y pueda ser accedido, en este caso, desde un browser en un dispositivo PC o móvil, ya que a pesar de que no será desarrollada una app móvil, la responsividad con la que la aplicación está siendo diseñada le permite ser accedida desde un browser en cualquier tipo de dispositivo.

Los contenedores que alojan la aplicación son señalados en la Imagen 3 con flecha roja, en la medida que la aplicación requiera ser escalada simplemente deben ser agregados más contenedores con la imagen de la aplicación para que esa escale de una forma horizontal. Los demás contenedores que se aprecian en la Imagen 3 corresponden a los de otros clientes que utilizan el mismo servicio de nube. La idea es la misma del servicio de transporte marítimo donde muchos contenedores de diversos clientes son transportados

en el mismo barco, en este caso ejecutados en las mismas máquinas virtuales que ejecutan el servicio de dockers/kubernetes.

Logo	Instaya			
	Programar Recogida	Rastrear	Cotizar	Ingresar
<div><div>¿Quienes somos?</div><div>Somos una empresa de mensajería que presta el servicio de transporte de paquetes entre las capitales de departamento de Colombia.</div><div>Entrega garantizada en: 24, 48 y 72 horas en función de la distancia.</div></div> <div></div>				
Informacion de contacto				

Logo	Instaya
<div><div>Inicio de sesion</div><div><div>Usuario</div><div><input type="text"/></div></div><div><div>Contraseña</div><div><input type="password"/></div></div><div>INGRESAR</div></div>	
Informacion de contacto	

## Tabla 2. Definición de artefactos

Product Backlog					
Nombre de la tarea	Detalles	Prioridad	Estado	Puntos de historia	Asignado al Sprint
Registro	Cómo <b>cliente quiero</b> ingresar a la plataforma de InstaYA, <b>registrarme</b> , con mis datos de contacto y personales, <b>para acceder</b> a los servicios de la plataforma y posteriormente agendar mi entrega	ALTA	En Progreso	3	SI
Hora de Entrega	Cómo <b>cliente quiero</b> ver las horas de entrega disponibles <b>para seleccionar</b> a mi conveniencia la que se ajuste a mis necesidades.	MEDIA	En Progreso	2	SI
Crear Orden	Cómo <b>cliente quiero</b> crear mi orden según el día, hora, las dimensiones del paquete, dirección de recogida, mis datos personales y los datos de entrega como la dirección, ciudad, identificación y nombre de quien recibe <b>para</b> que el envío sea realizado y entregado adecuadamente.	ALTA	En progreso	3	SI
Seguimiento a los Pedidos	Cómo <b>cliente quiero</b> hacer seguimiento a mis pedidos y ver el listado de estos con el estado actual del envío <b>para asegurarme</b> que los pedidos se estén entregando según los requerimientos (ejemplo: máximo en <b>24 horas</b> )	ALTA	Sin Empezar	3	SI
Actualizar Datos	Cómo <b>cliente quiero</b> actualizar los datos de mis órdenes realizadas con anterioridad <b>para poder</b> gestionirlas en caso de que algún dato haya cambiado o se haya digitado erróneamente.	BAJA	En Progreso	2	SI

**Prioridad:** Alta, Media, Baja

**Estado:** Sin Empezar - en Progreso - Finalizado

**Puntos de Historia:** Los puntos de historia representan un valor que sólo será relevante para el equipo de scrum, y que se usará para estimar el tamaño de una tarea (también llamada ítem, en inglés) en el backlog. Permite al equipo determinar la complejidad de la tarea. En este ejemplo calificaremos del 1-5 siendo 1 poca complejidad y 5 complejidad máxima.

**Asignado al Sprint:** SI/NO

## Backlog Sprint 1

User story	Descripción	Estimación	Responsable
Roles del Equipo	Durante esta fase se definen los roles de cada desarrollador según sus habilidades y fortalezas, sean tanto para el Back-End, Front-End, etc.	4 horas	Equipo Desarrollador
Diagrama de despliegue	Se definen los componentes, hardware, software, y herramientas a utilizar	10 horas	Equipo Desarrollador
Definición de Artefactos	Se hará una descripción completa o detallada de cada backlog con orden de prioridad, tiempo estimado y el responsable	5 horas	Equipo Desarrollador
Definición del Cronograma	Evaluar las fechas de entrega de cada sprint, tiempo que se le dedicada a cada tarea, además, del cronograma de entregas	8	Equipo Desarrollador
Entrega Sprint	Entregar el sprint en el tiempo estipulado	4	Equipo Desarrollador

## Backlog Sprint 2

User story	Descripción	Estimación	Responsable
Mockups	En esta fase se crea un documento con los Mockups y un documento con el Mapa de Navegabilidad	24 horas	Equipo de Desarrollo
Crear proyecto de Front-End en React	Diseño de la vista de login y de registro de usuario nuevo. Creación de la ventana del login, con estilos de Bootstrap, y del registro de nuevo usuario con los botones y links de acceso a cada uno, etc.	40 horas	Equipo de Desarrollo
Plantilla Bootstrap	Selección de plantilla e Implementación de la plantilla en el proyecto	4 horas	Equipo de Desarrollo
Creación de proyecto en GIT	Se crea el repositorio en Git, donde se cargará el proyecto	4 horas	Equipo de Desarrollo



### Backlog Sprint 3

User story	Descripción	Estimación	Responsable
Definición de componentes	Se crea un Documento con componentes definidos y el Esqueleto de componentes dentro del proyecto	36 horas	Equipo de Desarrollo
Creación de formularios	Creación de los formularios y vínculos teniendo en cuenta el mapa de navegabilidad y se Implementa el renderizado	40 horas	Equipo de Desarrollo
Simulación de solicitudes API	Se crea el documento descriptivo que permita elaborar pruebas de estas solicitudes	36 horas	Equipo de Desarrollo
Creación de base de datos no relacional	Se crean las DB Pedidos, DB Envíos y DB Actualización & Registro	10 horas	Equipo de Desarrollo
Creación del proyecto de Node.js		8 horas	Equipo de Desarrollo
Desarrollo del Back-End de la aplicación	Se define el listado de dependencias de la aplicación. Creación de módulos (funcionalidades) y Creación de módulos de autenticación y autorización de usuarios	40 horas	Equipo de Desarrollo
Implementación de conectividad con la base de datos	Desarrollo del API Rest con método Get para obtener los datos del usuario a buscar, conexión con base de datos y realización de sentencia para búsqueda de la información.	10 horas	Equipo de Desarrollo

### Backlog Sprint 4

User story	Descripción	Estimación	Responsable
Definición de requerimientos para el despliegue de la aplicación	Crear documento descriptivo con requerimientos mínimos para ejecutar la aplicación y dependencias	3 horas	Equipo de Desarrollo
Despliegue de la aplicación	Realizar el despliegue de la aplicación sobre un contenedor	12 horas	Equipo de Desarrollo

## Cronograma de tareas

[illegible]