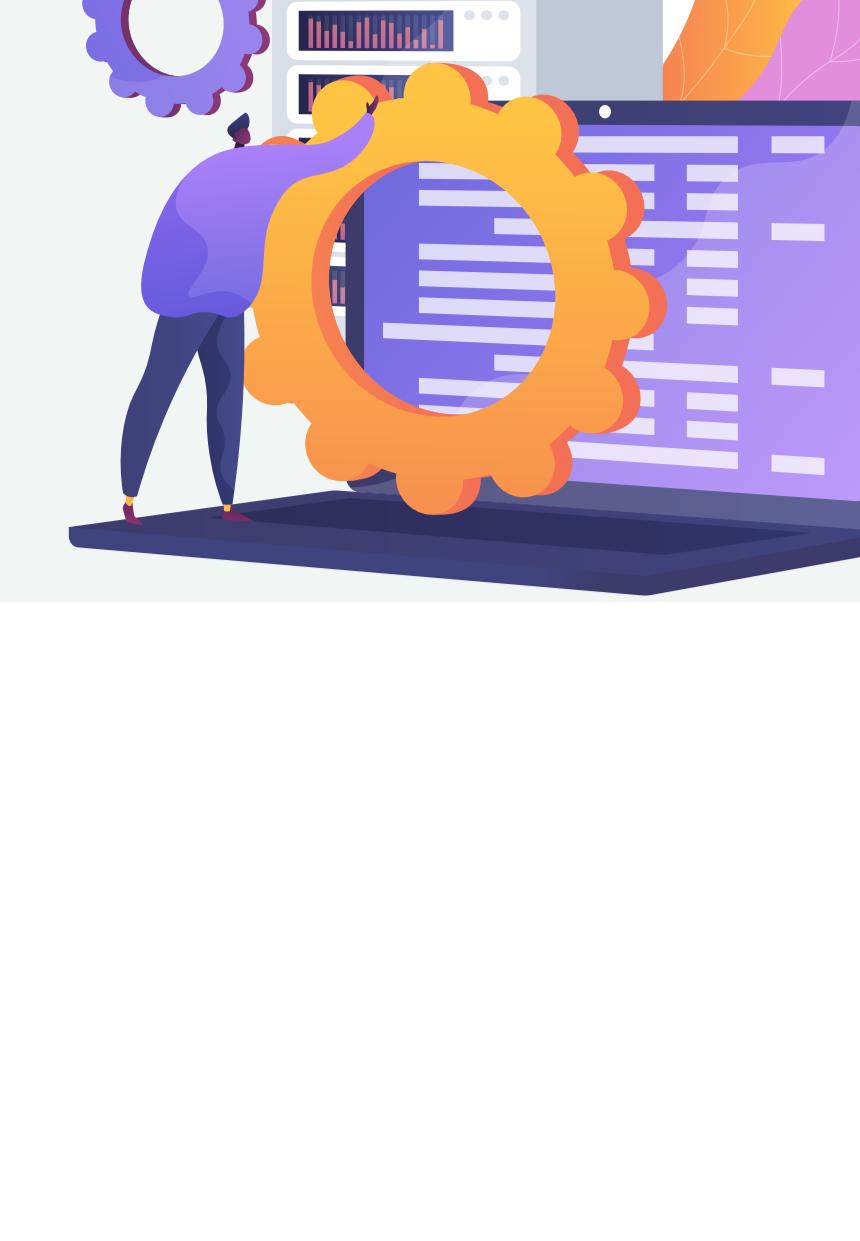




MACIÓN

SOFTWARE

Proyecto:



encuentras, de
manera ágil el

se deberán registrar los avances, las dificultades y el cumplimiento de compromisos de cada uno de los integrantes.

El tutor registrará el seguimiento de los avances que haga el equipo, e igualmente las dificultades y el cumplimiento de los compromisos de cada uno de sus miembros según lo registrado en los **Dailys** y en el administrador de proyectos (**Trello**). Si dentro de ese seguimiento se tiene alguna consulta o se requiere aclarar dudas e inquietudes, se podrá solicitar al tutor un encuentro sincrónico por medio de la plataforma; dicho encuentro quedará grabado en plataformas para su

2. ¿Cuál es el proyecto a desarrollar?

Planteamiento de la situación problema a desarrollar

Se propone analizar, diseñar y construir una aplicación de software que permita realizar el seguimiento de las ventas de un producto y/o servicio en una empresa (cada equipo podrá acordar con el tutor el tipo de productos y/o servicios que

3. ¿Qué debo alcanzar para desarrollar la aplicación web?

- Módulo administrador de productos

para el registro de procedimientos y para insertar, modificar y cada uno debe contar con los siguientes atributos: Identificador (Inmutable) descripción, valor unitario y estado (disponible).

- Módulo administrador de ventas. El sistema tendrá una interfaz gráfica para el registro de las ventas y otra para listar, buscar y actualizar las ventas realizadas (Actualizar se refiere a establecer los diferentes estados de la venta: En proceso, cancelada o entregada, o editar alguno de sus otros campos modificables). Cada venta debe contar con los siguientes atributos: Identificador único de venta (Inmutable), el valor total de la venta, identificador, cantidad y precio unitario de cada producto, fecha de venta, el documento de identificación y nombre del

(vendedor).

y el estado del usuario (pendiente/autorizado/no autorizado)

en este documento, pero se da libertad para que el equipo de desarrollo pueda acordar con el tutor cualquier funcionalidad adicional.

4. La palabra usuario es usada en el contexto del proyecto como la persona cualquiera que se registra en su aplicativo web, esta puede ser alguien que no tiene relación con la empresa, por lo que por defecto el usuario tiene estado pendiente y no tiene rol, dicha persona solo podrá ingresar a la aplicación exitosamente cuando el administrador le cambie el estado a autorizado y le

exitosamente cuando el administrador le cambie el estado a autorizado y asigne un rol. El administrador tiene acceso total a la aplicación, mientras que el vendedor únicamente a las interfaces de usuario correspondientes a la registro y maestro de ventas.

Para llevar a cabo este proyecto deberás tener en cuenta una serie de **indicaciones**, como verás a continuación.

rotarse los demás roles).

The diagram illustrates the responsibilities of four project roles, each associated with a specific task:

- 1. Desarrollador**: Generar las fuentes para el frontend y el backend
- 2. Analista**: Levantar las necesidades del negocio para convertirlas en requerimientos del sistema
- 3. Product Owner**: Conocer el producto *software* que se va a elaborar en el proyecto
- 4. Administrador de BD**: Gestionar y disponer el motor de base de datos para el proyecto de

The diagram consists of two rounded rectangular boxes. The left box is light blue and contains the text "5. Scrum Master". The right box is light green and contains the text "Gestionar todas las ceremonias y asignar los recursos a cada reunión".

4. Arquitectura de la solución propuesta para el proyecto final.

```
graph LR; subgraph API_REST [API REST]; direction TB; NodeJS[Node.js™]; Plus[+]; express[express]; end; subgraph FRONT_END [FRONT END]; React[React]; end; API_REST -- "Devuelve obj JSON" --> FRONT_END; FRONT_END -- "Llamado API/SRA" --> API_REST;
```

The diagram illustrates the proposed solution architecture. It features two main components: **API REST** and **FRONT END**. The **API REST** component, represented by a red-bordered box containing the Node.js logo and the word **express**, interacts with the **FRONT END** component, represented by a red-bordered box containing the React logo. A blue arrow labeled **Devuelve obj JSON** points from the API REST to the FRONT END, indicating the flow of data. Another blue arrow labeled **Llamado API/SRA** points from the FRONT END back to the API REST, indicating the reverse flow of requests.

The banner consists of two main parts. On the left, the Node.js logo is displayed, which features the word "node" in lowercase where each letter is a different icon: "n" is a house, "o" is a green hexagon, "d" is a dark grey hexagon with a small circle inside, and "e" is a dark grey hexagon with a dot inside. To the right of the "e" is the "JS" logo, which is "JS" inside a green hexagon with a trademark symbol. On the far right, there is a large, stylized blue atom or particle logo with three orbits.

5. ¿Cuál es la metodología que se utilizará para llevar a cabo el proyecto?

El desarrollo del proyecto se abordará mediante metodologías ágiles, usando el marco de trabajo **Scrum** (figura 1), consistente en un proceso que se aplica de manera regular y frecuente a las buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto.

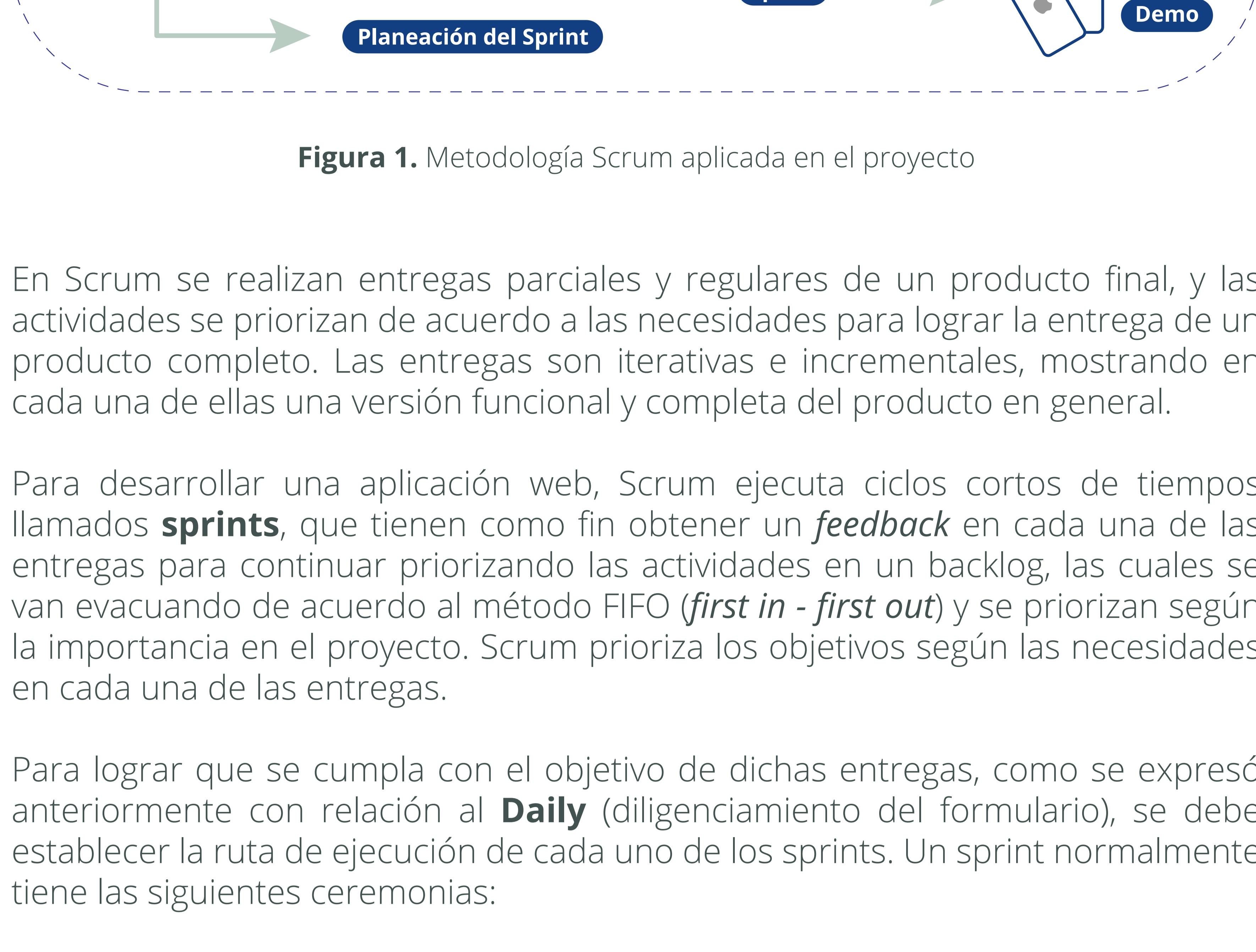


Figura 1. Metodología Scrum aplicada en el proyecto

En Scrum se realizan entregas parciales y regulares de un producto final, y las actividades se priorizan de acuerdo a las necesidades para lograr la entrega de un producto completo. Las entregas son iterativas e incrementales, mostrando en cada una de ellas una versión funcional y completa del producto en general.

Para desarrollar una aplicación web, Scrum ejecuta ciclos cortos de tiempos llamados **sprints**, que tienen como fin obtener un **feedback** en cada una de las entregas para continuar priorizando las actividades en un backlog, las cuales se van evacuando de acuerdo al método FIFO (*first in - first out*) y se priorizan según la importancia en el proyecto. Scrum prioriza los objetivos según las necesidades en cada una de las entregas.

Para lograr que se cumpla con el objetivo de dichas entregas, como se expresó anteriormente con relación al **Daily** (diligenciamiento del formulario), se debe establecer la ruta de ejecución de cada uno de los sprints. Un sprint normalmente tiene las siguientes ceremonias:

- **Sprint Planning**, que se realiza al comenzar el sprint.
- **Daily Scrum** (conocido comúnmente solo como la **Daily**), que es una reunión diaria de 15 minutos en la que deben participar todos los integrantes del equipo. En esta reunión, cada una de las personas del equipo responden las siguientes preguntas:
 - o ¿Qué hice ayer para contribuir al Sprint Goal?
 - o ¿Qué voy a hacer hoy para contribuir al Sprint Goal?
 - o ¿Tengo algún impedimento que me impida entregar?

Dicha reunión tiene como objetivo diligenciar el formulario con las preguntas que se presentaron anteriormente, y además se debe hacer el seguimiento y la administración al proyecto a través del **software Trello**, en el cual se llevará el registro de las metas determinadas en los sprints según las historias de usuario con cada uno de los responsables.

- **Sprint Review**, que es la reunión que ocurre al final del sprint. Es organizada por el producto **Owner** y se revisa cuál es la situación de todas las historias (**Backlog**) y sus nuevas condiciones para priorizar el trabajo posterior.

Para cumplir con dicha ceremonia, todos los días deberán reunirse los cinco integrantes del equipo durante 15 minutos, utilizando cualquier plataforma virtual para el encuentro. Dicha reunión tiene como objetivo diligenciar el formulario con las preguntas que se presentaron anteriormente, y además se debe hacer el seguimiento y la administración al proyecto a través del **software Trello**, en el cual se llevará el registro de las metas determinadas en los sprints según las historias de usuario con cada uno de los responsables.

En cada uno de los cinco sprints se deben desarrollar las actividades planteadas. De igual manera, el equipo debe cumplir con los compromisos estipulados correspondientes al sprint en la semana asignada. Esta entrega recibirá una nota que será el soporte para la certificación, pero, aunque el trabajo sea grupal, la nota será de manera individual. Debes tener presente que los sprints tienen un valor de **100%** de la calificación total del curso.

6. ¿Cuáles son las herramientas que se emplearán para desarrollar el proyecto?

Para lograr llevar a cabo el proyecto es necesario contar con un grupo de herramientas que permitirán organizar, planear, estructurar y desarrollar la aplicación web.

- **IDE** (interface development environment). En el ciclo 3 se utilizará Visual Studio Code para desarrollar frameworks como ReactJs y Nodejs.
- **Git**. Es un **software** de control de versiones de código distribuido, que se instala en la máquina local del desarrollador.

- **Github (repositorio)**. Es una carpeta para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

- **Trello (administrador de proyecto)**. Es un **software** de administración de proyectos que permite llevar el registro de todas las metas determinadas para cada uno de los sprints, así como las actividades de su responsable.

- **Heroku**. Es una plataforma en la nube que permite a las empresas construir, entregar y supervisar aplicaciones para luego poder alojarlas en la nube y posibilitar el acceso a los usuarios de las aplicaciones que se construyen.

Además, el lenguaje que se puede usar es JavaScript, y el stack tecnológico a emplear será:

- o React JS
- o HTML 5
- o CSS3
- o Lenguaje JavaScript
- o Lenguaje TypeScript
- o Node Js
- o Plataformas de desarrollo para Backend (Node JS)
- o Stack MERN
- o (OAuth2)
- o (**Heroku**)
- o (Git / GitHub)

Segunda indicación. Luego de conformado el equipo, se deben abordar los sprints, los cuales contienen las indicaciones necesarias para el desarrollo de la aplicación web. Cada sprint se detalla en un documento que encontrarás en la plataforma y que deberás ejecutar para poder alcanzar los objetivos propuestos por cada integrante. En la tabla 2 se presenta un resumen de lo que contiene cada sprint y el porcentaje de evaluación para cada uno de ellos.

Tabla 2.

Semana de entrega	Sprint	¿Qué se desarrollará?	Porcentaje de evaluación
Semana (2)	Sprint 1 Creación de un equipo de trabajo Definición de los roles de los integrantes Creación de un repositorio en Github Clic aquí	<ul style="list-style-type: none">• Crear un equipo de trabajo y asignar los roles de cada uno de los integrantes según la metodología Scrum• Definir las ceremonias de Scrum y el calendario de cada una• Tener creado el repositorio de las fuentes en GitHub• Realizar pull/request con cada cambio al proyecto• Tener creadas las ramas principales Development, Release, Master• Dar permisos a todos los integrantes del equipo• Hacer comentarios sobre cada cambio que se sube al repositorio <p>Todos los integrantes realizan cambios a las fuentes y se debe ver reflejado en el historial</p>	20%
Semanas (3 y 4)	Sprint 2 Creación de interfaces Desarrollo de interfaces gráficas mediante Reactjs Codificación mediante TypeScript y JavaScript HTML5, CSS3 Clic aquí	<ul style="list-style-type: none">• Crear la interfaz de usuario que permita autenticarse con un tercero en este caso OAuth 2 (no es necesario solicitar usuario y contraseña)• Crear la interfaz para el registro de los productos (identificador del producto, valor unitario, y estado: disponible, no disponible; no se contemplan impuestos ni valores adicionales) y crear la interfaz del maestro de productos (se pueden ver, buscar y actualizar los productos)• Crear la interfaz para el registro de las ventas (identificador de la venta, valor total de la venta, identificador, cantidad, y precio unitario de cada producto, fecha de venta, documento de identificación y nombre del cliente, y además deberá contar con un encargado de gestionar dicha venta, es decir, vendedor) y crear la interfaz de usuario para el maestro de las ventas (se puede listar, buscar y actualizar las ventas, esto quiere decir que se le puede cambiar el estado: en proceso, cancelada y entregada o editar algún campo modificable)• Crear la interfaz de usuario del maestro de usuarios para ver y actualizar el rol (administrador y vendedor) y el estado del usuario (pendiente / autorizado / no autorizado).	20%
Semana (5)	Sprint 3 Integración Backend y FrontEnd Clic aquí	<ul style="list-style-type: none">• Integrar las interfaces gráficas con Nodejs que a su vez tendrá que integrarse con la base de datos para completar la funcionalidad de gestionar la información (registrar, listar, actualizar) de los productos y usuarios.	20%
Semana (6)	Sprint 4 Seguridad Clic aquí	<ul style="list-style-type: none">• Implementar la autenticación y la autorización con OAuth2 de la aplicación para mejorar la seguridad del sitio web.• Integrar las interfaces gráficas con Nodejs que a su vez tendrá que integrarse con la base de datos para completar la funcionalidad de gestionar la información (registrar, listar, actualizar) de las ventas.	20%
Semana (7)	Sprint 5 Despliegue de la aplicación Clic aquí	<ul style="list-style-type: none">• Desplegar la aplicación en plataformas PaaS como HEROKU, entre otros	20%

Product backlog (Historias de usuario)

Uno de los recursos que se emplean dentro de la metodología Scrum para organizar la información es la historia de usuarios, la cual busca brindar la información de manera corta y simple. A continuación, te presentamos un esquema de la historia de usuario que debes manejar para el proyecto. También puedes profundizar un poco, a través del siguiente enlace, sobre la manera como se emplea esta herramienta:

<https://scrum.mx/informe/historias-de-usuario>

Feature: como analista/desarrollador requiero crear todo el entorno necesario para el manejo de la Gestión de la configuración, tanto en el equipo local como en el repositorio GitHub, para administrar todos los artefactos que se generan durante el proceso de desarrollo de software.

Historia de usuario: HU_001

Dado una herramienta para administrar las fuentes/artefactos en la nube

Cuando necesite crear un repositorio con mi propia cuenta

Entonces dispondré de un lugar en GitHub para almacenar la información/artefactos/fuentes de los proyectos de desarrollo

Historia de usuario: HU_002

Dado que tengo un repositorio de GitHub en la nube

Cuando necesite subir artefactos

Entonces podré tener artefactos ordenados en GitHub con la información del proyecto

Historia de usuario: HU_003

Dado que tengo instalado la aplicación Git en la máquina local

Cuando requiera enlazar un directorio con un repositorio en GitHub

Entonces podré subir nuevos artefactos al repositorio en GitHub

Historia de usuario: HU_004

Dado que tengo instalado la aplicación Git en la máquina local

Cuando requiera enlazar un directorio con un repositorio en GitHub

Entonces podré subir artefactos actualizados al repositorio en GitHub

Feature: Como vendedor o administrador necesito administrar la información de las ventas para gestionar la información del día a día.

Historia de usuario: HU_005

Como usuario

Dado que voy a ingresar al sistema de gestión de ventas

Cuando necesite validar mis credenciales

Entonces podré ver las opciones para ingresar mediante Gmail

Historia de usuario: HU_006

Como vendedor o administrador

Dado que ingreso al sistema de ventas

Cuando requiera listar la información de las ventas

Entonces podré ver la información de las ventas realizadas

Historia de usuario: HU_009

Como vendedor o administrador

Dado que ingreso al sistema de ventas

Cuando requiera verificar o actualizar la información de las ventas registradas en el sistema

Entonces podré realizar una búsqueda mediante el identificador de la venta, documento de identidad del cliente o nombre del cliente en la información de las ventas registradas en el sistema.

Historia de usuario: HU_010

Como vendedor o administrador

Dado que ingreso al sistema de ventas

Cuando requiera actualizar la información de las ventas

Entonces podré editar la información de la venta que deseo actualizar (Menos el identificador de la venta).

Historia de usuario: HU_011

Como vendedor o administrador

Dado que ingreso al sistema de ventas

Cuando actualice una venta

Entonces podré ver que se almacenó correctamente

Historia de usuario: HU_012

Como administrador

Dado que ingreso al sistema de ventas

Cuando necesite registrar un producto

Entonces podré ingresar los datos relacionados a un producto

Historia de usuario: HU_013

Como administrador

Dado que ingreso al sistema de ventas

Cuando registre un producto

Entonces podré ver que se almacenó correctamente

Historia de usuario: HU_014

Como administrador

Dado que ingreso al sistema de ventas

Cuando requiera listar la información de los productos

Entonces podré ver la información de los productos

Historia de usuario: HU_015

Como administrador

Dado que ingreso al sistema de ventas

Cuando requiera verificar o actualizar la información de los productos registrados en el sistema

Entonces podré realizar una búsqueda mediante el identificador del producto o descripción del producto en la información de los productos registrados en el sistema.

Historia de usuario: HU_016

Como administrador

Dado que ingreso al sistema de ventas

Cuando requiera actualizar la información de los productos

Entonces podré editar la información del producto que deseo actualizar (Menos el identificador del producto)

Historia de usuario: HU_017

Como administrador

Dado que ingreso al sistema de información de ventas

Cuando requiera actualizar la información de los usuarios del sistema

Entonces podré editar la información de los usuarios del sistema que deseo actualizar (Estado y rol)

Historia de usuario: HU_020

Como administrador

Dado que ingreso al sistema de ventas

Cuando actualice el estado o rol de un usuario

Entonces podré ver que se almacenó correctamente

Para llevar a cabo el seguimiento del proceso de desarrollo vamos a utilizar dos herramientas: **Trello** y **Github**. En Trello vamos a llevar un control de las tareas asignadas a cada miembro del grupo. Se recomienda crear las diferentes tareas para completar las historias de usuario, de tal manera que, toda la historia realizada de manera individual permita obtener la completitud de la historia de usuario.

Se deben crear cuatro columnas Backlog, ToDo, Progress y Done (figura 2).

- **Backlog.** En esta columna se encuentran las tareas a realizar, en general, de todo el proyecto; se obtienen al dividir cada historia de usuario en tareas.
- **ToDo.** Aquí van las tareas que se van a desarrollar en el sprint correspondiente.
- **Progress.** Aquí van las tareas que se están desarrollando en el momento; deben tener un responsable y un label que indique el sprint.
- **Done.** En esta columna se mueven las tareas que ya se terminaron.

Figura 2. Tareas en Trello

Github Es un sitio web en la nube que permite hospedar el código de las aplicaciones de cualquier desarrollador, y usa el protocolo Git. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y puedan colaborar con el desarrollo. Estas herramientas nos permiten tener procesos bien definidos para gestionar todos los cambios suscitados a lo largo del proyecto, y mediante el repositorio GitHub controlar e identificar los artefactos para llevar la trazabilidad de los mismos, mediante la adecuada información de las características del cambio que se presenta y poder devolverse en caso de error hacia el estado del componente anterior, permitiendo así saber quiénes están involucrados.

Para llevar a cabo el seguimiento del proceso de desarrollo vamos a utilizar dos herramientas: **Trello** y **Github**. En Trello vamos a llevar un control de las tareas asignadas a cada miembro del grupo. Se recomienda crear las diferentes tareas para completar las historias de usuario, de tal manera que, toda la historia realizada de manera individual permita obtener la completitud de la historia de usuario.

Se deben crear cuatro columnas Backlog, ToDo, Progress y Done (figura 2).

En Trello se crean cuatro columnas principales: Backlog, ToDo, Progress y Done. El Backlog contiene las tareas pendientes de desarrollo. El ToDo contiene las tareas que se están trabajando en el momento. El Progress indica el progreso de las tareas en desarrollo. El Done indica las tareas que ya han sido finalizadas.

En GitHub, cada repositorio tiene una página principal que muestra información sobre el código fuente, las拉 issues, los拉 pull requests y las拉 releases. Los commits muestran los cambios realizados en el código, incluyendo autores, fechas y descripciones detalladas. Los issues y los pull requests permiten a los desarrolladores comunicarse y colaborar en la mejora del proyecto.

Los repositorios en GitHub también ofrecen estadísticas detalladas como el lenguaje de programación predominante, el porcentaje de cobertura del código y las métricas de rendimiento. Esto proporciona una visión completa del estado actual del proyecto y facilita la identificación de problemas o áreas de mejora.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.

Al combinar Trello y GitHub, los equipos de desarrollo pueden mantener un seguimiento efectivo de las tareas y las mejoras, asegurando que tanto las tareas individuales como las historias de usuario se completen de manera integral y eficiente.