

## **Guía de Vertical y Proyecto - Simulación Laboral Tech**

Duración de la Simulación = cinco semanas.

Demo Day = Semana 4 ( Fecha a confirmar )

### **Principios de Simulación**

1. Aprender a trabajar en equipo.
2. Desarrollar tus soft skills.
3. Ganar experiencia real.
4. Evidenciar tu valor.

## **Vertical Tecnológica: WebApp**

### Descripción

La vertical de desarrollo de aplicaciones web se enfoca en la creación de plataformas y sistemas basados en la web que ofrecen funcionalidades y servicios a través de navegadores de Internet. Este campo abarca una amplia gama de proyectos, desde sitios web estáticos hasta aplicaciones web dinámicas y complejas. Los proyectos de aplicaciones web suelen requerir un enfoque multidisciplinario que combine el diseño de interfaces de usuario, el desarrollo de frontend y backend, así como la gestión de bases de datos y la seguridad web.

### Objetivo de Proyecto

El objetivo del proyecto de desarrollo de aplicaciones web es crear una plataforma en línea funcional y atractiva que cumpla con las necesidades y expectativas de los usuarios finales.

### **Tecnologías y tendencias**

Algunas de las tecnologías y tendencias más relevantes en el desarrollo de aplicaciones web incluyen:

- Frameworks de frontend: React.js, Angular, Vue.js, Vercel.
- Frameworks de backend: Node.js, Express.js, Django, Ruby on Rails, Flask.
- Bases de Datos: MySQL, PostgreSQL, MongoDB, Firebase.
- Herramientas de Desarrollo: Visual Studio Code, Git, GitHub, npm, Yarn.

El uso de estas tecnologías y frameworks está en constante evolución, y los profesionales en el campo deben mantenerse actualizados con las últimas tendencias para seguir siendo competitivos en el mercado laboral.

## Roles

- **Desarrollador Full-Stack:** Responsable de trabajar en todas las capas de una aplicación web, desde el frontend hasta el backend.
- **Diseñador de Experiencia de Usuario (UX):** Diseña interfaces de usuario intuitivas y atractivas para mejorar la experiencia del usuario.
- **Desarrollador Frontend:** Se especializa en la creación y optimización de la interfaz de usuario.
- **Desarrollador Backend:** Responsable de la lógica y funcionalidad detrás de una aplicación web, gestionando servidores, bases de datos y la lógica del lado del servidor.
- **Especialista DevOps:** En el rol de DevOps, jugarás un papel crucial en la automatización y gestión de la infraestructura. Tu objetivo es asegurar una colaboración fluida entre el desarrollo y las operaciones de TI para un proceso de desarrollo eficiente.
- **Project Manager:** Estarás a cargo de guiar y coordinar a tu equipo a lo largo de la simulación. Te asegurarás de que cada miembro comprenda su rol, que la comunicación fluya entre todos los miembros y que el proyecto avance según lo planificado.

## Colaboración Multidisciplinaria

Es crucial resaltar la importancia del trabajo conjunto entre diferentes roles, lenguajes de programación y tecnologías. Aquí hay algunos ejemplos de colaboración:

- Back-End y Front End: El desarrollador backend crea las API necesarias para que el frontend pueda interactuar con el servidor. Por ejemplo, el backend proporciona endpoints para autenticación, almacenamiento de datos y recuperación de información, mientras que el frontend consume estos endpoints para mostrar la interfaz de usuario y enviar datos al servidor.
- Front-end y Diseñador UX/UI: El diseñador UX/UI colabora estrechamente con el desarrollador frontend para crear una experiencia de usuario intuitiva y atractiva. Por ejemplo, el diseñador UX/UI puede proporcionar diseños de

interfaz de usuario que el desarrollador frontend implementa utilizando HTML, CSS y JavaScript.

- Backend y Tester: El desarrollador backend trabaja con el tester para garantizar la funcionalidad y la calidad del software. Por ejemplo, el tester puede escribir pruebas de integración para asegurarse de que los diferentes componentes del backend funcionen correctamente juntos, mientras que el desarrollador backend corrige los errores identificados por el tester.
- Project Manager y Back-end: El Project Manager trabaja en estrecha colaboración con el desarrollador backend para establecer los objetivos del proyecto, asignar tareas y garantizar que se cumplan los plazos. Por ejemplo, el Project Manager puede priorizar las funciones del backend en función de las necesidades del cliente y coordinar con el desarrollador backend para asegurarse de que se completen a tiempo.
- Java Back-end y Angular Front-end: El desarrollador backend en Java proporciona la lógica empresarial y la capa de acceso a datos utilizando tecnologías como Spring Framework, mientras que el desarrollador frontend en Angular utiliza TypeScript y Angular para crear una interfaz de usuario dinámica que se conecta con el backend a través de APIs RESTful.
- C# Back-end y React Front-end: El desarrollador backend en C# utiliza ASP.NET Core para crear servicios web RESTful, mientras que el desarrollador frontend en React utiliza JavaScript y React para crear componentes reutilizables y construir una interfaz de usuario interactiva que se comunica con el backend a través de solicitudes HTTP.

#### Algunas Tecnologías y Herramientas más utilizadas

- Frontend: HTML, CSS, JavaScript, React.js, Angular, Vue.js.
- Backend: Node.js, Express.js, Django, Ruby on Rails, Flask.
- Base de Datos: MySQL, PostgreSQL, MongoDB, Firebase.
- Herramientas de Desarrollo: Visual Studio Code, Git, GitHub, npm, Yarn.

#### Distribución de Tareas:

El propio equipo, con la asistencia y moderación del team líder, se encargará de la distribución de tareas y responsabilidades, promoviendo la autonomía y la toma de decisiones colaborativas. Se debe buscar consenso en las decisiones.

### Reuniones de Equipo

La comunicación efectiva es clave para el éxito del proyecto. Coordinen horarios de reuniones diarias y semanales considerando las diferentes zonas horarias de los miembros. Algunas pautas:

- Flexibilidad Horaria: Adapten los horarios a las diferentes ubicaciones geográficas y preferencias individuales.
- Coordinación Anticipada: Planifiquen con anticipación y usen herramientas para encontrar el mejor momento para todos.
- Rotación Equitativa: Distribuyan equitativamente los horarios para no afectar constantemente a los mismos miembros.
- Comunicación Clara: Informen claramente sobre los horarios y cualquier cambio utilizando canales de comunicación efectivos.
- Respeto Mutuo: Respeten las diferencias de zona horaria y estén dispuestos a comprometerse para encontrar soluciones que funcionen para todos.

### Transparencia en los conocimientos

La transparencia en los conocimientos ayuda a crear un ambiente de confianza y colaboración en el equipo, donde todos se sienten cómodos compartiendo sus conocimientos y aprendiendo unos de otros. Esto contribuye a un mejor rendimiento del equipo y al éxito general del proyecto.

### **Tecnologías a utilizar**

Es importante que utilicen aquellas tecnologías que ustedes sepan como así también otras que no conozcan para incorporar al Stack para enriquecer su proyecto.

Consideren tener un buen equilibrio entre lo que ya saben y lo que no.

Proponemos desde No Country un balance 70/30:

- 70% tecnologías que el equipo haya utilizado previamente.
- 30% tecnologías que el equipo no conozca.

La idea es mantener una buena proporción entre ambas partes para pulir y consolidar los conocimientos que ya tenemos y seguir dedicando tiempo a incorporar nuevas tecnologías.

## Funcionalidades y User Stories

En este proyecto, es fundamental comprender y definir las funcionalidades del sistema a través de User Stories, lo que proporciona un marco de pensamiento claro y centrado en el usuario. Aquí tienes una descripción de cada concepto:

- **Funcionalidades:** Son las características y capacidades específicas que tiene el sistema o la aplicación. Estas funcionalidades son las acciones que el sistema puede realizar para cumplir con los requisitos del usuario y satisfacer sus necesidades.
- **User Stories (Historias de Usuario):** Son descripciones breves de una funcionalidad del sistema desde la perspectiva del usuario final. Las User Stories se enfocan en lo que el usuario quiere lograr y por qué, sin entrar en detalles técnicos. Suelen seguir el formato "Como [tipo de usuario], quiero [realizar una acción], para [obtener un beneficio]".
- **Tipos de Usuario:** Representan diferentes roles o perfiles de usuarios que interactúan con el sistema. Cada tipo de usuario tiene necesidades y objetivos específicos. Algunos ejemplos: Usuario | Administrador | Cliente | Visitante

Al comprender las funcionalidades, las User Stories y los diferentes tipos de usuarios, se puede desarrollar un sistema que satisfaga las necesidades y expectativas de los usuarios de manera efectiva y eficiente. Esto ayuda a mantener el enfoque en la experiencia del usuario y garantizar que el proyecto cumpla con sus objetivos principales.

## Recordatorios

- Asistir a las 2 reuniones semanales con su Team Leader.
- Sprint Semanal con su Team Leader
- Crear repositorio de proyecto en nuestra organización de [Github](#).
- **Fecha clave** = Demo Day - Semana 4

## Rol de No Country en el Proyecto

No Country ejercerá el rol de "cliente", esto tiene como intención simular el proceso de entorno laboral real donde el equipo debe realizar una demostración al cliente. En este contexto cada Team Leader y los diferentes canales de Slack de la comunidad representan los intereses del cliente.

## Marco de Trabajo

El Marco de Trabajo se fundamenta en dos conceptos clave: MVP (Producto Mínimo Viable) y los principios de Lean Startup.

Producto Mínimo Viable (MVP) es la versión más básica de un producto que satisface las necesidades fundamentales del usuario. Adoptar esta mentalidad es crucial para enfocarse en lo esencial y evitar el desperdicio de recursos.

Los principios de Lean Startup promueven la experimentación rápida y el aprendizaje validado. Esto implica estar preparados para enfrentar desafíos y conflictos, resaltando la importancia de la flexibilidad y la adaptabilidad como soft skills fundamentales para el éxito en proyectos tecnológicos.

## Presentación de Proyecto - Demo Day

Durante esta etapa, se alienta a los equipos a utilizar técnicas de storytelling para comunicar de manera clara y atractiva la visión, el propósito y el impacto de su proyecto. Esto implica identificar y resaltar los puntos clave, las soluciones innovadoras y el valor añadido que aporta el producto al usuario final.

El Demo Day es el evento culminante donde los equipos tienen la oportunidad de exponer sus proyectos ante la comunidad. **Nuestra recomendación es que tengan como meta presentar el deploy de su mvp.** Durante este día, se espera que los equipos demuestren su capacidad para comunicar de manera efectiva la propuesta de valor de su proyecto, respondiendo preguntas y recibiendo feedback de manera constructiva.