



Proyecto Tutoría UTULAB

Solicitante: I.T.S - Instituto Tecnológico Superior Arias Balparda.

Nombre Fantasía, de la nueva empresa: Neuro Code Technologies

Grupo: 3ro MN

Turno: Nocturno

Nombres de los integrantes del grupo: Federico Albistur, Gabriel Jorge, Rodrigo Machado, Federico Sciapicony

Fecha de entrega: 28/07/2025

Instituto Tecnológico Superior Arias Balparda.

**Blvr. José Batlle y Ordóñez 3570 esq. Gral. Flores –
Montevideo.**



ÍNDICE

1. Presentación.....	4
1.1 Introducción.....	4
1.2 Creación del Nombre.....	4
1.3 Creación del Logo.....	5
2. Paradigma de Organización de Equipo de Proyecto.....	6
2.1 Paradigma seleccionado: Organización en Células Autónomas - Método Agile/Scrum.....	6
2.2 ¿En qué consiste este paradigma?.....	6
2.3 Principales roles en Scrum:.....	7
2.4 Fundamentación de la elección.....	7
2.5 Aplicación práctica en nuestro equipo.....	7
3. Roles asignados al equipo.....	9
• Rodrigo Machado – Scrum Master / Líder del Proyecto, Responsable de Documentación Técnica y Desarrollador de Scripts	9
• Federico Albistur – Product Owner, Responsable de Ingeniería de Software y Redactor de Actas.....	10
• Federico Sciapicony – Frontend y Backend Developer, Responsable del DER y Documentación de Base de Datos.....	10
• Gabriel Jorge – Backend Developer y Responsable de Documentación Técnica de Programación.....	11
4. Esquema de funcionamiento de los roles (Organigrama).....	12
4.1 Fundamentación del organigrama.....	12
5. Fundamentación de la elección de los roles dentro del proyecto..	13
6. Estructura de toma de decisiones y plan de acción en casos críticos.....	14
6.1 Toma de decisiones operativas (día a día).....	14
6.2 Toma de decisiones estratégicas o de impacto.....	14
6.3 Plan de acción ante casos críticos.....	15
6.4 Comunicación y transparencia.....	15
7. Reglas del grupo – Neuro Code Technologies.....	16
8. Esquema de Sanciones.....	18
9. Herramienta de Planificación Seleccionada.....	20

10. Documento de Inicio y Planificación.....	20
11. Ciclo de vida.....	21
11.1 Shashimi (Cascada e Incremental).....	21
12. Comparativa de ciclos de vida.....	23
12.1 Justificación de la Elección del Modelo Shashimi.....	25
13. Tablero de planificación.....	26
Bibliografía.....	28



1. Presentación

1.1 Introducción

En el marco del proyecto final, desarrollamos una propuesta empresarial innovadora: **Neuro Code Technologies**, una empresa dedicada al desarrollo de software personalizado, moderno y eficiente.

El objetivo principal de la empresa es brindar soluciones tecnológicas a medida, utilizando lenguajes de programación actuales, metodologías ágiles y herramientas de inteligencia artificial. Nos enfocamos en transformar ideas en productos digitales funcionales, atendiendo las necesidades específicas de empresas, emprendimientos y profesionales.

Nuestros servicios incluyen el diseño y desarrollo de aplicaciones web, móviles y de escritorio, así como la integración de bases de datos, automatización de procesos y consultoría tecnológica.

A través de un enfoque creativo y técnico, buscamos impulsar la transformación digital en diversos sectores, garantizando calidad, innovación y seguridad en cada proyecto.

1.2 Creación del Nombre

El nombre **Neuro Code Technologies** fue creado a partir de la fusión de tres conceptos clave:

- **Neuro:** Inspirado en las redes neuronales artificiales, representa la inteligencia, el aprendizaje y la capacidad de adaptación tecnológica.
- **Code:** El código es la base de toda solución digital, lo que conecta la lógica con la funcionalidad.
- **Technologies:** Representa la amplitud del proyecto, que abarca múltiples disciplinas del ámbito tecnológico.

Esta combinación da como resultado una identidad sólida, moderna y alineada con los avances actuales en desarrollo e innovación.

1.3 Creación del Logo

El nuevo logo refleja de manera visual y simbólica la identidad de **Neuro Code Technologies**. Está compuesto por los siguientes elementos:

- **Hexágono naranja:** Representa la estructura, la tecnología y la energía. Su forma alude a la organización y al orden lógico, fundamentales en el mundo de la programación.
- **Red neuronal azul:** Representa la inteligencia artificial, los datos y la capacidad de procesamiento de la empresa. Simboliza la conexión entre puntos de información, tal como lo hacen los sistemas informáticos complejos.
- **Elemento circular tipo chip:** Enfatiza la precisión, la innovación y el enfoque digital. A su vez, destaca el eje tecnológico del proyecto.
- **Tipografía moderna:** El nombre de la empresa se muestra con un tipo de letra técnica y geométrica, transmitiendo profesionalismo y claridad.
- **Eslogan:** "Transformando ideas en tecnología" refuerza la visión de la empresa como un puente entre la creatividad y la implementación digital.

Los colores naranja, azul y negro fueron elegidos estratégicamente:

- **Naranja:** Representa energía, creatividad y entusiasmo.
- **Azul:** Transmite confianza, profesionalismo y tecnología.
- **Negro:** Aporta elegancia, sobriedad y solidez.



2. Paradigma de Organización de Equipo de Proyecto

2.1 Paradigma seleccionado: Organización en Células Autónomas - Método Agile/Scrum

Para el desarrollo de nuestro proyecto dentro de la empresa **Neuro Code Technologies**, hemos optado por implementar el **paradigma de organización en células autónomas**, basado en metodologías ágiles, particularmente en el marco de trabajo "Scrum". Este paradigma se alinea con la naturaleza del desarrollo de software moderno, que requiere adaptabilidad, colaboración constante y entregas progresivas.

2.2 ¿En qué consiste este paradigma?

Este enfoque propone trabajar en **equipos reducidos y multidisciplinarios** donde no existen jerarquías rígidas, sino **roles definidos y responsabilidades compartidas**. Cada miembro del equipo tiene voz activa en la toma de decisiones y participa en todas las fases del desarrollo.

El equipo se organiza en torno a iteraciones cortas llamadas **"sprints"**, al final de las cuales se entrega una parte funcional del producto. Durante todo el proceso, se realizan reuniones periódicas para evaluar avances, identificar obstáculos y adaptar el trabajo en función del aprendizaje continuo.

2.3 Principales roles en Scrum:

- **Product Owner:** Representa los intereses del cliente y define las prioridades del producto. Asegurar que el equipo trabaje en lo que realmente aporta valor.



- **Scrum Master:** Actúa como facilitador del proceso. Se encarga de que se sigan los principios ágiles y remueve los impedimentos que afecten al equipo.
- **Equipo de Desarrollo:** Grupo que diseña, programa, prueba e implementa las funcionalidades requeridas.

2.4 Fundamentación de la elección

Elegimos este paradigma por las siguientes razones:

Flexibilidad y adaptabilidad:

El entorno del desarrollo de software es dinámico. Las necesidades del cliente o los requerimientos pueden cambiar en cualquier momento. Este paradigma permite **adaptarse rápidamente** sin perder el rumbo del proyecto.

Participación activa y compromiso del equipo:

Todos los integrantes del grupo participan en las decisiones clave. Esto genera un mayor sentido de pertenencia y motivación, ya que cada miembro se siente parte importante del producto final.

Mejora continua:

Cada sprint finaliza con una reunión de retrospectiva donde el equipo reflexiona sobre lo que funcionó y lo que puede mejorar. Este ciclo de aprendizaje **optimiza los procesos de forma constante**.

Visibilidad y transparencia:

A través de reuniones diarias informales (dailies) y tableros de trabajo, se mantiene una **comunicación abierta** sobre el estado del proyecto, lo que facilita la organización y la anticipación de problemas.

Enfoque en la entrega de valor:

Cada sprint tiene como objetivo entregar algo que funcione, aunque sea parcial. Esto permite **validar avances reales** y priorizar las funciones más importantes desde el principio.

2.5 Aplicación práctica en nuestro equipo



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



Durante el desarrollo del proyecto, dividimos nuestro trabajo en **módulos funcionales** (como backend, frontend, base de datos y pruebas). En cada sprint asignamos tareas específicas, definimos objetivos claros y nos reunimos periódicamente para evaluar el progreso.

Adoptamos herramientas colaborativas (GitHub Projects) para organizar y visualizar las tareas, y coordinamos nuestras acciones mediante breves reuniones al inicio de cada jornada, donde cada integrante comenta:

- Qué hizo la reunión anterior.
- Qué hará para la próxima reunión.
- Si tiene algún bloqueo o dificultad.

Este enfoque nos permitió trabajar **de forma ordenada, ágil y colaborativa**, maximizando nuestras fortalezas y respetando los tiempos de cada integrante del grupo.



3. Roles asignados al equipo

Para la organización del equipo de trabajo se optó por una estructura basada en el paradigma **Agile/Scrum**, ajustada a un equipo reducido y multidisciplinario. Esta forma de trabajo permite que cada integrante asuma **roles múltiples**, combinando responsabilidades técnicas, de documentación y de gestión.

A continuación, se describen los roles y tareas asignadas a cada miembro del equipo:

- **Rodrigo Machado – Scrum Master / Líder del Proyecto, Responsable de Documentación Técnica y Desarrollador de Scripts**

- Coordina y supervisa todas las fases del proyecto.
- Facilita la metodología de trabajo Scrum (planificación, dailies, retrospectivas).
- Se encarga de la **documentación de UTULAB y Sistemas Operativos**.
- Desarrolla el **script bash** para la configuración del servidor.
- Supervisa entregas y cumplimiento de los objetivos.
- Participa activamente en las pruebas funcionales del sistema (tester).



- **Federico Albistur – Product Owner, Responsable de Ingeniería de Software y Redactor de Actas**

- Define y documenta los **requerimientos del sistema**, tanto funcionales como no funcionales.
- Se encarga de la **documentación de Ingeniería de Software**.
- Colabora con en la redacción de la **documentación de UTULAB**.
- Redacta las **actas formales** del equipo, dejando constancia de decisiones y reuniones.
- Supervisa que la implementación técnica cumpla con los requerimientos definidos.
- Participa activamente en el testing.

- **Federico Sciapicony – Frontend y Backend Developer, Responsable del DER y Documentación de Base de Datos**

- Encargado del **frontend** del sistema (interfaz de usuario y experiencia visual).
- Colabora en el **backend** utilizando **PHP y manejo de datos en formato JSON**.
- Diseñó el **DER (Diagrama Entidad-Relación)** de la base de datos, definiendo su estructura lógica.
- Participa en la **documentación de la base de datos** y el funcionamiento general del sistema.



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



→ Involucrado en el testing de componentes e interacción cliente-servidor.

- **Gabriel Jorge – Backend Developer y Responsable de Documentación Técnica de Programación**

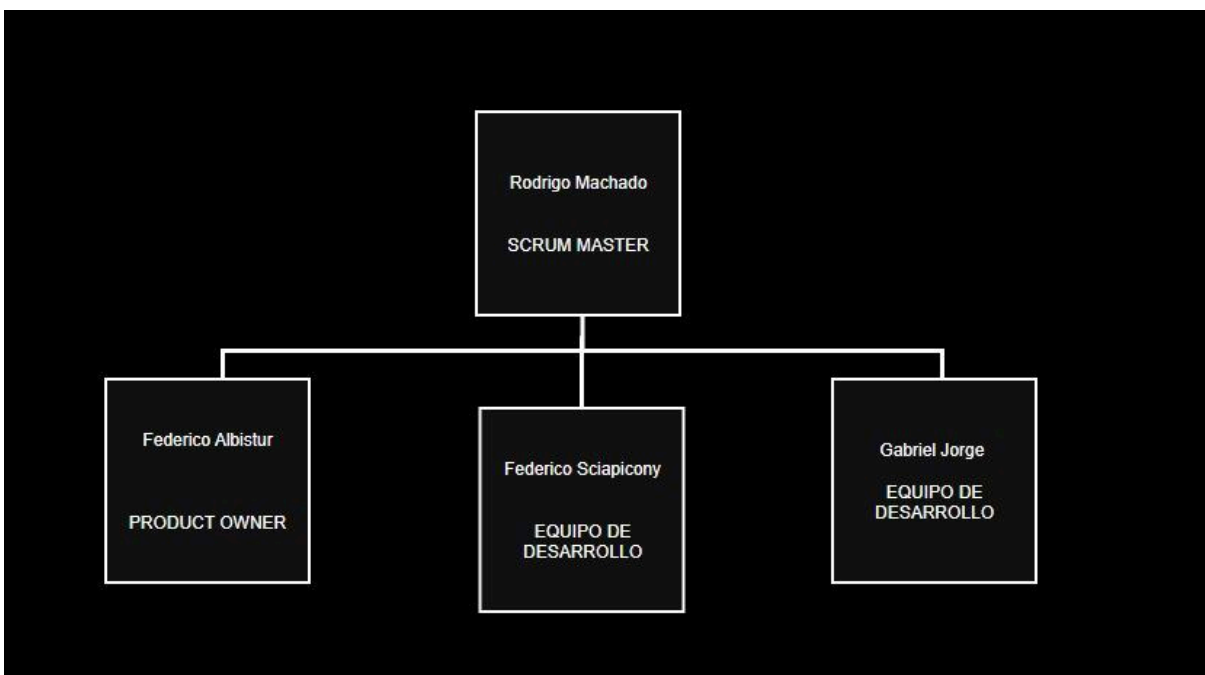
→ Desarrolla el **backend de la aplicación utilizando PHP**.

→ Diseña y gestiona la **base de datos**.

→ Redacta la documentación correspondiente a **Programación Full Stack**.

→ Participa en pruebas de funcionamiento y testing de errores.

4. Esquema de funcionamiento de los roles (Organigrama)



4.1 Fundamentación del organigrama

El organigrama utilizado es de tipo **jerárquico horizontal**, adaptado al enfoque ágil (Scrum) aplicado por el equipo. En él, se representa al **Scrum Master** (líder del proyecto) como figura central de coordinación, mientras que los demás integrantes asumen roles técnicos específicos, conectados de manera directa sin niveles jerárquicos rígidos.

Esta estructura permite una **asignación clara de responsabilidades**, pero al mismo tiempo mantiene la **colaboración transversal** entre los miembros. Todos participan en tareas clave como documentación, programación y testing, promoviendo la **flexibilidad**,



autonomía y trabajo en equipo, pilares fundamentales en los proyectos ágiles.

5. Fundamentación de la elección de los roles dentro del proyecto

La asignación de roles dentro del equipo se definió a partir de un **análisis FODA grupal**, que permitió identificar las **fortalezas, debilidades, oportunidades y amenazas** de cada integrante en relación con el proyecto. Esta evaluación fue clave para organizar el trabajo de manera eficiente, aprovechando al máximo las capacidades individuales.

Como resultado de este análisis, se designó a un **Scrum Master (líder del proyecto)** con perfil organizativo y técnico, encargado de la coordinación general y la documentación principal. Las áreas de **desarrollo frontend, backend, base de datos y documentación específica** se distribuyeron considerando los conocimientos previos, el nivel de experiencia y el interés particular de cada miembro.

Además, se asignaron tareas analíticas y de registro formal (como actas y documentación de requerimientos) a quienes demostraron habilidades en redacción técnica y orden documental.

Finalmente, todos los integrantes participan en el proceso de **testing**, promoviendo la mejora continua y asegurando la calidad del producto.

Este enfoque permitió definir una estructura equilibrada, funcional y coherente con los principios del trabajo ágil, optimizando el rendimiento del equipo y asegurando el cumplimiento de los objetivos del proyecto.



6. Estructura de toma de decisiones y plan de acción en casos críticos

Dentro del marco organizativo del equipo y siguiendo los principios de la metodología ágil (Scrum), se definió una estructura de toma de decisiones que garantiza **agilidad, consenso y respuesta efectiva** ante situaciones críticas o imprevistos.

6.1 Toma de decisiones operativas (día a día)

- Las decisiones menores o técnicas se resuelven de forma **colaborativa** en las reuniones informales breves de seguimiento (dailies).
- Cada integrante propone soluciones y se elige la más adecuada por **consenso rápido**, priorizando el avance fluido del proyecto.

6.2 Toma de decisiones estratégicas o de impacto

- En casos de decisiones importantes (como cambios de requerimientos, rediseño, retrasos críticos o problemas técnicos mayores), se convoca una **reunión extraordinaria del grupo (reunión formal)**.
- Se analiza el problema, se presentan alternativas y se elige la solución **por consenso o mayoría simple**.
- El **Scrum Master / Líder del proyecto** se encarga de moderar la discusión, evaluar la viabilidad de las opciones y coordinar la ejecución de la decisión tomada.

6.3 Plan de acción ante casos críticos

En caso de presentarse situaciones imprevistas que afecten directamente al avance del proyecto (errores graves, pérdida de datos, inasistencias prolongadas, fallas técnicas, etc.), se aplicará el siguiente plan:

1. **Detección y registro** del problema por cualquier miembro del equipo.
2. **Evaluación rápida** en conjunto: ¿qué impacto tiene y cuánto retrasa el proyecto?
3. **Propuesta de soluciones** por parte del equipo.
4. **Toma de decisión consensuada**, priorizando la solución más realista y menos riesgosa.
5. **Redefinición de tareas o cronograma**, si es necesario.
6. **Seguimiento activo** por parte del Scrum Master hasta la resolución completa.
7. **Aplicación de sanción, si corresponde**, en caso de que el problema haya sido ocasionado por negligencia, reiterado incumplimiento o falta de compromiso por parte de un integrante. La sanción será definida por consenso del grupo y registrada formalmente en el acta correspondiente.

6.4 Comunicación y transparencia

- Todas las decisiones importantes se registran en las **actas formales**, a cargo de **Federico Albistur**.
- Esto asegura **transparencia**, seguimiento de acuerdos y trazabilidad en caso de revisiones posteriores.



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



7. Reglas del grupo – Neuro Code

Technologies

1. **Participación activa:**

Todos los integrantes deben contribuir activamente en reuniones, tareas y decisiones del grupo.

2. **Asistencia obligatoria:**

La asistencia a reuniones internas es obligatoria. Se debe justificar cualquier inasistencia con al menos 12 horas de anticipación, salvo casos de urgencia.

3. **Cumplimiento de tareas:**

Las tareas asignadas deben cumplirse en el tiempo acordado y con el nivel de calidad esperado.

4. **Comunicación fluida:**

Se deben responder los mensajes grupales dentro de las 24 h y estar disponibles en los canales oficiales (WhatsApp, etc.).

5. **Respeto mutuo:**

Se exige un trato respetuoso entre todos los miembros, promoviendo un ambiente de trabajo colaborativo.

6. **Puntualidad:**

Las entregas deben realizarse con puntualidad y las reuniones comienzan a la hora establecida (con una tolerancia de 15 minutos).



7. **Uso responsable de herramientas:**

Se debe mantener actualizado el repositorio Git y registrar los avances individuales de forma clara.

8. **Documentación:**

Todos los acuerdos y tareas deben quedar registrados en actas de reunión formales.

9. **Toma de decisiones:**

Las decisiones se tomarán por mayoría simple. En caso de empate, definirá el Líder de Proyecto.

10. **Manejo de conflictos:**

Los desacuerdos deberán resolverse primero entre los implicados de forma directa. De persistir, se llevará a reunión con el líder para resolución formal.

11. **Uso de actas y revisiones:**

Las actas deben compartirse al finalizar cada reunión. Los integrantes tienen 24 horas para solicitar correcciones o agregar observaciones.

8. Esquema de Sanciones

Código	Falta	Sanción
F1	Inasistencia no justificada a una reunión	Advertencia verbal y registro en el acta
F2	Entrega tardía sin justificación	Pérdida de derecho a voto en la siguiente reunión
F3	Reiteración de F1 o F2 (entre 2 y 5 veces)	Redistribución de tareas y revisión de rol
F4	Reiteración de F1 (más de 5 veces)	Pérdida de derecho a voto hasta haberse puesto al día en todas sus obligaciones no realizadas
F5	No responder mensajes importantes en plazo	Advertencia y posible reasignación de tareas urgentes
F6	Falta de respeto o lenguaje inapropiado	Reunión interna de revisión y suspensión temporal
F7	No cumplir con tareas asignadas sin justificación	Exposición del caso en reunión y posible redistribución



F8	Incumplimiento en el uso del repositorio Git	Advertencia y capacitación obligatoria
F9	Modificar archivos sin aviso ni control de versiones	Reversión del cambio y advertencia formal
F10	Rechazo sistemático al trabajo en grupo / desinterés evidente	Registro formal y reunión de evaluación
F11	Incumplimiento sistemático de reglas (6 o más faltas graves)	Evaluación de permanencia en el equipo y posible exclusión definitiva



9. Herramienta de Planificación Seleccionada

Para la gestión del proyecto en **Neuro Code Technologies**, hemos seleccionado una combinación ágil de herramientas y metodologías que aseguran eficiencia y valor.

GitHub Projects es nuestra herramienta central para la planificación y el seguimiento. La elegimos por su integración nativa con GitHub, que vincula directamente tareas y código, facilitando la trazabilidad. Su flexibilidad con tableros Kanban y su transparencia mejoran la colaboración y la visualización del progreso en tiempo real. Además, permite automatizaciones básicas para optimizar el flujo de trabajo.

Este es nuestro repositorio en GitHub:

<https://github.com/Mowaaa/NeuroCodeProyecto>

10. Documento de Inicio y Planificación

La planificación en Neuro Code Technologies se formaliza a través de los siguientes elementos clave:

- **Visión del Producto:** Documento de alto nivel que define la misión y objetivos estratégicos del software, guiando el proyecto.
- **Backlog del Producto:** Lista dinámica y priorizada de todas las funcionalidades y requisitos, gestionada continuamente en **GitHub Projects**.
- **Planificación de Sprints:** Reuniones al inicio de cada *sprint* para seleccionar y detallar las tareas a realizar, creando el *Sprint Backlog* también en **GitHub Projects**.



- **Actas de Reunión:** Registros formales de decisiones y acuerdos clave para mantener la transparencia y trazabilidad.
- **Documentación Técnica y Funcional:** Creación y actualización constante de toda la documentación del sistema (requisitos, diseño, base de datos, programación, scripts), a cargo de los roles específicos del equipo.

11. Ciclo de vida

11.1 Shashimi (Cascada e Incremental)

En **Neuro Code Technologies**, nuestro ciclo de vida de desarrollo se basa en el **Modelo Shashimi**, que conceptualizamos como una **fusión directa de los modelos en cascada e incremental**. Esta aproximación es el pilar de nuestra filosofía para construir software de manera eficiente, adaptable y entregando valor de forma continua.

El **Modelo Shashimi** se distingue por ser un enfoque **iterativo e incremental** donde cada porción de trabajo que abordamos es una **pieza funcional y terminada en sí misma**.

Esto se traduce en las siguientes características fundamentales de nuestro ciclo de vida:

- **Es Incremental en su Naturaleza:** El desarrollo se realiza en **iteraciones cortas (sprints)**, entregando el software en **pequeñas porciones funcionales**. En lugar de esperar a que el producto completo esté terminado, construimos y liberamos incrementos de valor de manera sucesiva, permitiendo una evolución constante del producto.



- **Es Cascada dentro de Cada Incremento:** La disciplina de las fases del **modelo en cascada** (análisis, diseño, codificación y pruebas) se aplica **dentro de cada una de esas pequeñas porciones o sprints**. Esto significa que cada funcionalidad específica atraviesa por su propio ciclo completo de estas etapas antes de ser considerada "terminada" para ese incremento.
- **Solapamiento Intencional de Etapas:** La característica distintiva del Shashimi es el **solapamiento de estas actividades** dentro de cada iteración. Las fases de análisis, diseño, codificación y pruebas no se ejecutan de manera estrictamente secuencial para todo el proyecto; en su lugar, se realizan de forma **concurrente o superpuesta** en el mismo periodo de trabajo. Por ejemplo, mientras se está programando una parte, ya se están probando funcionalidades del incremento anterior y diseñando las del siguiente, todo enfocado en la "rebanada" actual.
- **Entrega Continua de Valor Funcional:** Al final de cada *sprint*, nuestro equipo produce un **incremento de producto que es funcional y potencialmente entregable**. Este enfoque permite a los interesados ver y validar el progreso de forma temprana y frecuente, asegurando que el desarrollo siempre se alinee con las necesidades del negocio.

Al basar nuestro ciclo de vida en el Modelo Shashimi como esta combinación de cascada e incremental, Neuro Code Technologies asegura una **integración continua** de los componentes del software, una **retroalimentación temprana** y una **reducción proactiva de riesgos**. Esto nos permite adaptarnos ágilmente a los cambios, mantener una alta calidad en el producto y garantizar que cada esfuerzo contribuya directamente a un resultado tangible y valioso.



12. Comparativa de ciclos de vida

Característica / Modelo	Cascada	Incremental	Iterativo	Espiral	Modelo en V	Modelo Shashimi
Enfoque Principal	Lineal, secuencial por fases completas.	Versiones sucesivas que añaden funcionalidad.	Ciclos repetitivos que refinan el producto.	Basado en análisis de riesgos y prototipos.	Extensión del modelo cascada, con fuerte enfoque en validación.	Rebanadas funcionales completas por sprint.
Flujo de Trabajo	Fase por fase, sin retrocesos.	Repetición de fases por incremento.	Ciclos de mejora continua.	Cuadrones de riesgo progresivos.	En forma de "V": fase de desarrollo y su prueba correspondiente.	Fases superpuestas por porción de valor.
Ideal Cuando	Requisitos estables.	Se requiere entrega temprana.	Requisitos cambiantes.	Proyectos complejos y de alto riesgo.	Alta necesidad de validación y verificación formal.	Requisitos dinámicos, foco en calidad y feedback



						tempran o.
Entrega de Valor	Al final del proyecto.	Cada cierto número de incrementos.	Frecuentemente, al final de cada iteración.	En cada prototipo.	Al final, después de una exhaustiva verificación.	En cada sprint.
Gestión de Cambios	Difícil y costosa.	Fácil en nuevas versiones.	Parte natural del proceso.	Gestión activa de riesgos y cambios.	Muy difícil, el cambio impacta en varias etapas.	Flexible, con cambios en cada iteración.
Detección de Riesgos	Tarde, al final.	Por incremento.	Desde el inicio.	Permanente.	Tarde, durante la validación final.	Desde el primer sprint.
Involucramiento del Cliente	Bajo.	Moderado.	Alto y constante.	Alto en validaciones.	Bajo, centrado en pruebas finales.	Alto y continuo.



Documentación	Extensa y formal.	Moderada.	Mínima necesaria.	Fuerte foco en riesgos.	Muy extensa, detallada por fase.	Enfocada por funcionalidad.
Ventajas Clave	Orden y claridad.	Entrega anticipada de valor.	Adaptabilidad, calidad.	Control del riesgo.	Alta calidad por validación y verificación paralela.	Equilibrio, flexibilidad, valor desde el inicio.
Desventajas Clave	Muy rígido.	Puede requerir retrabajo.	Exige organización y experiencia.	Costoso y complejo.	Muy rígido, costoso en cambios.	Requiere buena comunicación y equipo disciplinado.

12.1 Justificación de la Elección del Modelo Shashimi

Elegimos el **modelo Shashimi** porque combina lo mejor de los modelos **en cascada** e **incremental**. Este enfoque nos permite trabajar con **estructura dentro de cada iteración (sprint)**, manteniendo a la vez la **flexibilidad necesaria para adaptarnos** a cambios de requisitos, validaciones tempranas y entregas funcionales continuas.



ANEP



UTU

DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONAL



Instituto Tecnológico Superior
UTU

13. Tablero de planificación

🌐 Distribucion de tareas

📅 Backlog

👤 Capacidad de equipo

🔄 La Iteración actual

📄 Hoja de ruta

📁 Mis artículos

+ Nuevas vistas

🟢 **Todo** 1 / 5 Estimación: 0

Este artículo no ha sido iniciado.

🟢 NeuroCodeProyecto no 8

Prueba de Servidores/Terminales

P0

+ Añáda un elemento

🟡 **En progreso** 1 / 10 Estimación: 0

Esto se está trabajando activamente en

🟢 NeuroCodeProyecto 5

UTULAB

P1

+ Añáda un elemento

🟣 **Hecho** 6 Estimación: 0

Esto se ha completado.

🟢 NeuroCodeProyecto 2

Script Linux

P0 Iteración

🟢 NeuroCodeProyecto 3

BD de script

P1 Iteración

🟢 NeuroCodeProyecto 7

Software de Ingeniería

P1 Iteración

🟢 NeuroCodeProyecto no 6

Pasaje a tablas . Diccionario de Datos

P2

🟢 NeuroCodeProyecto 9

Documentacion S.O

Iteración

🟢 NeuroCodeProyecto no4

+ Añáda un elemento

Distribucion de tareas
Añadir actualización de estado

Backlog
 Capacidad de equipo
 La iteración actual
 Hoja de ruta
 Mis artículos
+ Nuevas vistas

Assignees ▾

- elplibbe 4
- Falbsitur 3
- FScIapIcony 3
- Mowaaa 3

Mostrar valores vacíos

Q. Filtra por palabra clave o por campo										Descartar	Guardar
Título	Situación	Tamaño	Estimación	Iteración							
P1 3 Estimación: 0 ...											
1 Script Linux .	Hecho			Iteración							
2 Prueba de Servidores/Terminales .	Todo										
+ artículo											
P1 3 Estimación: 0 ...											
3 UTULAB 5	En progreso										
4 BD de script 3	Hecho			Iteración							
5 Software de Ingeniería 7	Hecho			Iteración							
+ artículo											
P2 1 Estimación: 0 ...											
6 Pasaje a tablas . Diccionario de Datos 6	Hecho										
+ artículo											
Sin prioridad 2 Estimación: 0 ...											
7 Documentacion S.O 9	Hecho			Iteración							
8 Pasaje una tabla .	Hecho										
+ artículo											

Distribución de tareas

Backlog Capacidad de equipo La iteración actual **Hoja de ruta** Mis artículos + Nuevas vistas

Situación Deseleccionar

☒ **Todo**
 Este artículo no ha sido iniciado. 1

☐ **En progreso**
 Esto se está trabajando activamente en 1

☒ **Hecho**
 Esto se ha completado. 6

Mostrar valores vacíos

Filtro por palabra clave o por campo

Mar 2025	Apr 2025	20 de mayo de 2025	Jun 2025	Jul 2025	Ago 2025	Sep 2025	20 de octubre	Nov
10	24	7	21	5	19	2	16	30
14	28	1	15	29	12	26	9	23
18	1	15	29	12	26	9	23	30
22	5	19	2	16	30	14	28	11
26	9	23	6	20	3	17	31	14
30	13	27	10	24	7	21	19	5
3	17	31	14	28	11	25	8	22
7	21	4	18	1	15	29	12	26
11	25	8	22	5	19	2	16	30
15	29	12	26	9	23	17	31	14
19	1	15	29	12	26	9	23	30
23	5	19	2	16	30	14	28	11
27	9	23	6	20	3	17	31	14
31	13	27	10	24	7	21	19	5
3	17	31	14	28	11	25	8	22
7	21	4	18	1	15	29	12	26
11	25	8	22	5	19	2	16	30
15	29	12	26	9	23	17	31	14
19	1	15	29	12	26	9	23	30
23	5	19	2	16	30	14	28	11
27	9	23	6	20	3	17	31	14
31	13	27	10	24	7	21	19	5
3	17	31	14	28	11	25	8	22
7	21	4	18	1	15	29	12	26
11	25	8	22	5	19	2	16	30
15	29	12	26	9	23	17	31	14
19	1	15	29	12	26	9	23	30
23	5	19	2	16	30	14	28	11
27	9	23	6	20	3	17	31	14
31	13	27	10	24	7	21	19	5
3	17	31	14	28	11	25	8	22
7	21	4	18	1	15	29	12	26
11	25	8	22	5	19	2	16	30
15	29	12	26	9	23	17	31	14
19	1	15	29	12	26	9	23	30
23	5	19	2	16	30	14	28	11
27	9	23	6	20	3	17	31	14
31	13	27	10	24	7	21	19	5
3	17	31	14	28	11	25	8	22
7	21	4	18	1	15	29	12	26
11	25	8	22	5	19	2	16	30
15	29	12	26	9	23	17	31	14
19	1	15	29	12	26	9	23	30
23	5	19	2	16	30	14	28	11
27	9	23	6	20	3	17	31	14
31	13	27	10	24	7	21	19	5
3	17	31	14	28	11	25	8	22
7	21	4	18	1	15	29	12	26
11	25	8	22	5	19	2	16	

**ANEP****UTU**DIRECCIÓN GENERAL
DE EDUCACIÓN
TÉCNICO PROFESIONALInstituto Tecnológico Superior
UTU

Bibliografía

- **APD.** (2021, 14 de diciembre). *¿Cuáles son los roles de la metodología Scrum?* APD. <https://www.apd.es/roles-metodologia-scrum/> [APD](#)
- **Asana.** (2025, 15 de febrero). *Scrum: conceptos clave y cómo se aplica en la gestión de proyectos.* Asana. <https://asana.com/es/resources/what-is-scrum> [Asana](#)
- **Amazon Web Services (AWS).** (s. f.). *¿En qué consiste Scrum?* AWS. <https://aws.amazon.com/es/what-is/scrum/> [Amazon Web Services, Inc.](#)
- **ENAE International Business School.** (s. f.). *¿Qué es la metodología Scrum?* ENAE. <https://www.enaes.es/blog/metodologia-scrum> [enaes.es](#)
- **Grupo Aspasia.** (s. f.). *Qué es la Metodología SCRUM y características.* Grupo Aspasia. <https://grupoaspasia.com/es/glosario/metodologia-scrum/grupoaspasia.com>
- **¿Qué es Sashimi en el desarrollo software?** (2021, 4 de abril). 9 Geishas. <https://9geishas.com.ar/desarrollo-sashimi/>
- **Ciclo de Vida: Conceptos.** (2012, 19 de abril). Sings UFPS. <http://sings-ufps.blogspot.com/2012/04/ciclo-de-vida-conceptos.html>
- **Ciclo de vida Sashimi.** (s.f.). <https://www.google.com/search?q=ingsoftware.weebly.com>. <https://ingsoftware.weebly.com/ciclo-de-vida-sashimi.html>
- **Ciclo de vida Sashimi.** (s.f.). ingenieriadesoftwaretdea.weebly.com. <https://ingenieriadesoftwaretdea.weebly.com/ciclo-de-vida-sashimi.html>