



INGENIERÍA DE SISTEMAS DE SOFTWARE

CURSO: [1ASI0572 Desarrollo de Soluciones IoT](#)

ENUNCIADO DEL TRABAJO FINAL

PROFESORES:	León Baca, Marco Antonio Velásquez Núñez, Ángel Augusto Vera Olivera, David Carlos
SECCIONES:	Todas
FECHA DE EVALUACIÓN:	Semana 15
CICLO ACADEMICO:	2025-20

Objetivo:

El presente documento define el trabajo final y la rúbrica que permite evaluar el logro del curso [1ASI0572 Desarrollo de Soluciones IoT](#)

Logro del curso:

Al finalizar el curso, el estudiante comunica resultados y proceso de ingeniería aplicado para el ciclo de desarrollo y despliegue de una solución IoT distribuida bajo una arquitectura basada en IoT devices con Embedded Systems, Edge Computing y Cloud Computing, que considera el contexto real y satisface requisitos para empresas o público en general, con un enfoque innovador e inclusivo, en un ambiente ágil y colaborativo, haciendo uso de tecnologías open-source para distintos niveles de los componentes para la solución, considerando características para el proceso y visualización de información cuantitativa.

En Ingeniería de Software, el logro contribuye a alcanzar el:

ABET – EAC - Student Outcome 5: La capacidad de funcionar efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos.

Enunciado

[El curso de Desarrollo de Soluciones IoT tiene una naturaleza teórico práctica, por lo que es necesario evidenciar la capacidad para desarrollar soluciones basadas en IoT aplicando los conceptos, técnicas y buenas prácticas impartidos en el curso. Este trabajo tiene por objetivo desarrollar una solución de software compuesta por un conjunto de productos digitales, con la implementación de una arquitectura de](#)

aplicación empresarial basadas en tecnologías IoT que brinde soporte a un modelo de negocio independiente y sostenible. Las aplicaciones orientadas a usuarios deben contar con una interfaz adaptable a las dimensiones del dispositivo, e incluir las características que permitan satisfacer las necesidades identificadas para los segmentos objetivo, haciendo uso de las herramientas y tecnología utilizadas en el curso, aplicando conceptos y técnicas revisados en clase o resultado de la investigación. Como parte del conjunto de características, la solución debe considerar features para el proceso, cálculo en base a expresiones matemáticas y/o estadísticas y visualización de información cuantitativa recolectada a través de los dispositivos IoT.

Por otro lado, la arquitectura de la solución debe estar conformada por un RESTful API de elaboración interna, Web Applications y Mobile Applications integradas con dicho API, con una interfaz adaptable a las dimensiones de visualización de los dispositivos cliente, un Edge API de elaboración interna, así como Embedded Applications, integradas con el Edge API. Se incluye como parte del alcance un sitio web estático para el Landing Page del modelo de negocio. Como parte de las características a considerar, además de la integración con servicios RESTful de desarrollo interno, debe acceder a algún servicio externo de terceros. Es importante recalcar que la experiencia debe ser consistente entre el Landing Page y las aplicaciones Web, native Mobile u otro tipo de productos digitales con los que interactúe el usuario. Los call-to-action para cada segmento objetivo del Landing Page deben redirigir al usuario a las vistas correspondientes en las Web Applications o a los sitios de descarga de las aplicaciones móviles o a un punto de acceso adecuado según la naturaleza del producto digital referenciado. El alcance incluye adicionalmente un prototipo físico del dispositivo IoT, con un subconjunto relevante de características

Exposición

La exposición forma parte de la nota. Si al momento de la exposición el profesor determina que el alumno no ha hecho parte o la totalidad del trabajo debido a que el alumno no supo responder correctamente a las preguntas realizadas el profesor podrá considerar descontar puntos en funcionalidades ya implementadas del trabajo. La frase “En esa parte me ayudaron” no será considerada como válida por lo que el alumno deberá realizar el trabajo de forma total.

Instrucciones para la entrega del trabajo

Un representante del grupo subirá los archivos en la actividad indicada por el docente, los siguientes archivos: documento de Informe del Proyecto (en versión .PDF), documento de presentación (en versión PowerPoint y .PDF), Reporte de participación (en Microsoft Word y .PDF), archivo .zip con artefactos y proyectos de software (si fuera aplicable), video de exposición (como URL privado publicado en Microsoft Stream/Clipchamp y como archivo en formato .mp4).

Informe de proyecto

El informe de proyecto se elaborará de forma colaborativa durante todo el ciclo de vida del proyecto, en el formato Markdown (organizado en uno o mas archivos con extensión .md, siendo el archivo principal README.md) en un repositorio de control de versiones que forme parte de una organización pública de GitHub. Dicho

repositorio debe evidenciar mediante commits los aportes de los miembros del equipo. Debe aplicar GitFlow y conventional commits para su creación y evolución. A partir de dicho repositorio se generan las exportaciones en formato .pdf para las entregas (TB1, TP1, TB2, TF1).

Videos de exposición

Todo video de exposición debe mostrar a cada uno de los integrantes ante cámara explicando la construcción de los artefactos del trabajo, centrándose en mostrar los artefactos según la rubrica correspondiente al hito, apoyándose en una presentación de PowerPoint y demostrando los artefactos en las herramientas indicadas. La duración máxima del video es de 30 minutos.

Horarios de entrega

Para TB1 y TB2, la entrega se realiza hasta 24 horas después del horario programado regular de la sesión síncrona de la semana (en caso de dos sesiones síncronas semanales, se considera la segunda sesión síncrona).

Para TP1, la entrega se realiza 24 horas antes de la hora programada de la segunda sesión síncrona semanal.

Para TF1, la entrega se realiza 24 horas antes de la hora programada de la primera sesión síncrona semanal.

Nomenclatura de Archivos

Documento de Informe de Proyecto: Seguir la estructura de nombre *upc-pre-202520-1asi0572-<NRC>-<startup>-report-<tbn/tp1/tf1>* (.pdf)

Presentación (keynote) de Proyecto: Seguir la estructura de nombre *upc-pre-202520-1asi0572-<NRC>-<startup>-keynote-<tbn/tp1/tf1>* (.pptx y .pdf)

Documento de Informe de Participación: Seguir la estructura de nombre *upc-pre-202520-1asi0572-<NRC>-<startup>-performance-<tbn/tp1/tf1>* (.docx y .pdf)

Video de exposición: Seguir la estructura de nombre *upc-pre-202520-1asi0572-<NRC>-<startup>-expo-<tbn/tp1/tf1>* (.mp4)

Recomendaciones generales

Revisar con detenimiento los documentos Final Project Statement (este documento), así como las rúbricas.

Exposición

Para cada entrega, el equipo grabará en video su exposición con anticipación. El video contará de una edición que muestre la presentación de PowerPoint junto con la muestra en pantalla de artefactos como diagramas u otros que lo requieran, sincronizado con la explicación ante cámara de los participantes. En la primera parte de la exposición, debe incluirse tomas de cada participante hablando a la cámara y presentándose. Debe editarse para que sólo sea un video cuidando que el contenido no exceda el límite impuesto por Microsoft Stream/Clipchamp.

El enlace del video debe incluirse en el Document Report, dentro de un anexo titulado “Videos de Exposiciones” y especificando la entrega a la que corresponde la Exposición. Debe entregarse además el archivo de video (ver anexos).

Sustentación síncrona

Para TP1 y TF1, la sesión de clase en la que esté programada la entrega se enfocará en la sustentación de los proyectos. Cada equipo iniciará con una exposición planificada para una duración máxima de 12 minutos. Luego se realizará la

sustentación con preguntas a los participantes de forma indistinta, sobre diversos aspectos del proyecto previamente entregado y expuesto (vía video pre-grabado).

Participant Performance Report

El Participant Performance Report es un documento en Word que elabora el Team Leader, en el cual explica y califica el desempeño de cada uno de los miembros de su equipo (ver anexos). En este Documento el coordinador resume la participación de cada integrante y la asigna a cada uno una calificación entre 0 y 20 (siendo las alternativas de calificación 20, 16, 13, 07 o 00). Cada entrega debe incluir un *Participant Performance Report* sobre el desempeño de cada participante en relación con dicha entrega.

Final Project Keynote

Cada entrega incluye una presentación de PowerPoint cuyo contenido se relaciona con el ciclo de vida del proyecto, priorizando contenido relacionado con la entrega en cuestión. Entre las diapositivas deben incluir una de presentación del equipo con las fotos de los miembros de la startup, identificados por nombre, apellido y carrera.

Final Project Document Report

Tener cuidado con el respeto del Template en caso aplique.

User Stories, Product Backlog, Sprint Backlogs

De incluirse en la entrega artefactos con texto como los User Stories con Acceptance Criteria y Work-items, deben estar redactados en el Document Report, no como captura de imágenes de herramientas, como por ejemplo los Product Backlog en PivotalTracker y los Task boards en Trello.

Artefactos de UX (Cuando corresponda)

Cuando corresponda, cada equipo tendrá creado un proyecto en UXPressia, en el cual deben elaborar los artefactos de UX soportados por la herramienta. Capturas en imagen de estos artefactos deben incluirse en el Document Report en las secciones adecuadas y con las explicaciones y análisis que correspondan. Del mismo modo, los diagramas en herramientas indicadas como LucidChart o Figma /Adobe XD, deben incluirse como imágenes en el documento junto con su explicación, además de ser mostradas y explicadas en el video de exposición.

Estructura del Informe

Cada grupo debe entregar un informe detallando cada una de las secciones que se muestran a continuación:

Carátula

Logo de la Universidad

Nombre de Universidad, carrera, ciclo

Código del curso y Nombre del curso

NRC (Número de referencia de curso)

Nombre del profesor

"Informe de Trabajo Final"

Nombre del startup

Nombre del producto

Relación de integrantes (incluyendo para cada caso Código, Apellidos y Nombres)

Mes y año

Registro de Versiones del Informe

Project Report Collaboration Insights

Contenido

Tabla de contenidos

Student Outcome

Capítulo I: Introducción

- 1.1. *Startup Profile*
 - 1.1.1. Descripción de la Startup
 - 1.1.2. Perfiles de integrantes del equipo
- 1.2. *Solution Profile*
 - 1.2.1. Antecedentes y problemática
 - 1.2.2. Lean UX Process.
 - 1.2.2.1. Lean UX Problem Statements.
 - 1.2.2.2. Lean UX Assumptions.
 - 1.2.2.3. Lean UX Hypothesis Statements.
 - 1.2.2.4. Lean UX Canvas.
- 1.3. Segmentos objetivo.

Capítulo II: Requirements Elicitation & Analysis

- 2.1. Competidores.
 - 2.1.1. Análisis competitivo.
 - 2.1.2. Estrategias y tácticas frente a competidores.
- 2.2. Entrevistas.
 - 2.2.1. Diseño de entrevistas.
 - 2.2.2. Registro de entrevistas.
 - 2.2.3. Análisis de entrevistas.
- 2.3. Needfinding.
 - 2.3.1. User Personas.
 - 2.3.2. User Task Matrix.
 - 2.3.3. User Journey Mapping.
 - 2.3.4. Empathy Mapping.
- 2.4. Big Picture EventStorming.
- 2.5. Ubiquitous Language.

Capítulo III: Requirements Specification

- 3.1. User Stories.
- 3.2. Impact Mapping.
- 3.3. Product Backlog.

Capítulo IV: Solution Software Design

- 4.1. Strategic-Level Domain-Driven Design.

- 4.1.1. Design-Level EventStorming.
 - 4.1.1.1 Candidate Context Discovery.
 - 4.1.1.2 Domain Message Flows Modeling.
 - 4.1.1.3 Bounded Context Canvases.
- 4.1.2. Context Mapping.
- 4.1.3. Software Architecture.
 - 4.1.3.1. Software Architecture System Landscape Diagram.
 - 4.1.3.2. Software Architecture Context Level Diagrams.
 - 4.1.3.2. Software Architecture Container Level Diagrams.
 - 4.1.3.3. Software Architecture Deployment Diagrams.
- 4.2. Tactical-Level Domain-Driven Design
 - 4.2.X. Bounded Context: <Bounded Context Name>
 - 4.2.X.1. Domain Layer.
 - 4.2.X.2. Interface Layer.
 - 4.2.X.3. Application Layer.
 - 4.2.X.4. Infrastructure Layer.
 - 4.2.X.5. Bounded Context Software Architecture Component Level Diagrams.
 - 4.2.X.6. Bounded Context Software Architecture Code Level Diagrams.
 - 4.2.X.6.1. Bounded Context Domain Layer Class Diagrams.
 - 4.2.X.6.2. Bounded Context Database Design Diagram.

Capítulo V: Solution UI/UX Design

- 5.1. Style Guidelines.
 - 5.1.1. General Style Guidelines.
 - 5.1.2. Web, Mobile and IoT Style Guidelines.
- 5.2. Information Architecture.
 - 5.2.1. Organization Systems.
 - 5.2.2. Labeling Systems.
 - 5.2.3. SEO Tags and Meta Tags
 - 5.2.4. Searching Systems.
 - 5.2.5. Navigation Systems.
- 5.3. Landing Page UI Design.
 - 5.3.1. Landing Page Wireframe.
 - 5.3.2. Landing Page Mock-up.
- 5.4. Applications UX/UI Design.
 - 5.4.1. Applications Wireframes.
 - 5.4.2. Applications Wireflow Diagrams.
 - 5.4.2. Applications Mock-ups.
 - 5.4.3. Applications User Flow Diagrams.
- 5.5. Applications Prototyping.

Capítulo VI: Product Implementation, Validation & Deployment

- 6.1. Software Configuration Management.
 - 6.1.1. Software Development Environment Configuration.
 - 6.1.2. Source Code Management.
 - 6.1.3. Source Code Style Guide & Conventions.
 - 6.1.4. Software Deployment Configuration.

- 6.2. Landing Page, Services & Applications Implementation.
- 6.2.X. Sprint n
 - 6.2.X.1. Sprint Planning n.
 - 6.2.X.2. Aspect Leaders and Collaborators.
 - 6.2.X.3. Sprint Backlog n.
 - 6.2.X.4. Development Evidence for Sprint Review.
 - 6.2.X.5. Testing Suite Evidence for Sprint Review.
 - 6.2.X.6. Execution Evidence for Sprint Review.
 - 6.2.X.7. Services Documentation Evidence for Sprint Review.
 - 6.2.X.8. Software Deployment Evidence for Sprint Review.
 - 6.2.X.9. Team Collaboration Insights during Sprint.
- 6.3. Validation Interviews.
 - 6.3.1. Diseño de Entrevistas.
 - 6.3.2. Registro de Entrevistas.
 - 6.3.3. Evaluaciones según heurísticas.
- 6.4. Video About-the-Product.

Conclusiones

Conclusiones y recomendaciones.
Video About-the-Team.

Bibliografía

Anexos

Consideraciones sobre los componentes

Registro de Versiones del Informe

El objetivo de esta sección es resumir las modificaciones relevantes que se realizan al informe durante el ciclo de vida del proyecto.

Esta sección inicia en una página nueva y se incluye un cuadro con la siguiente estructura:

Versión	Fecha	Autor	Descripción de modificación

Como primera línea de la tabla se incluye la primera versión del informe. A partir de ello, se considera modificaciones relevantes la adición de secciones, eliminación de secciones, correcciones o mejoras producto de retroalimentación recibida del docente o producto de la autocrítica del equipo. Cada línea de versión debe incluir solo un autor. Esto quiere decir que, entre una entrega y otra, pueden irse generando varias versiones del informe. Todo ello debe quedar reflejado en este cuadro de Registro.

Project Report Collaboration Insights

En esta sección el equipo indica el URL del repositorio para el Project Report en la organización de GitHub del equipo. Adicionalmente, para cada entrega explica cómo

se han desarrollado las actividades de elaboración del informe y se presenta capturas en imagen de los analíticos de colaboración y commits en GitHub para el repositorio del informe, realizados por los miembros del equipo. Todos los miembros del equipo deben tener participación en la elaboración del informe. Esta sección debe ir expandiéndose con descripciones y evidencias en cada entrega. Lo descrito y evidenciado debe tener coherencia con el *Registro de Versiones del Informe*.

Contenido

La sección inicia en una nueva página. Para esta sección **utilice hipervínculos de Markdown para construir la tabla de contenido**. Considere en la elaboración 4 niveles de esquema. Recuerde actualizar y verificar la tabla de contenidos antes de cada entrega.

Student Outcome

Cada participante del equipo debe colaborar a fin de que se redacte como grupo los sustentos y evidencias de las actividades realizadas en el trabajo final han ayudado a desarrollar cómo las dimensiones del student outcome. Por ello en esta sección debe quedar descrito por escrito, la relación entre el outcome, sus dimensiones y el trabajo que han realizado. Esto se complementa con lo reflejado en los testimonios expuestos que forman parte del video *About The Team*.

La sección inicia en una nueva página. Debe incluir el párrafo introductorio y el cuadro de Student Outcome tal como se indica en la sección de Anexos de este documento. En las celdas Acciones realizadas, debe especificarse cada participante: Apellidos, Nombres y a continuación, cada entrega (TB1, TB2, etc.) con las acciones específicas realizadas que se relacionen con el criterio del Outcome al que corresponda la celda. Esta celda se irá expandiendo en cada entrega. Las celdas Conclusiones se llenan de forma grupal y son acumulables, es decir se van expandiendo en cada entrega. Vea especificaciones sobre cómo estructurar esta sección en el *Anexo A. Estructura para la sección Objetivo del Estudiante (Student Outcome)*. El párrafo introductorio antes del cuadro que se aprecia en el anexo referido debe colocarse también de forma idéntica en su informe.

Startup Profile

Incluye la descripción de startup¹, perfiles de los miembros del equipo, incluyendo foto de participante, nombres y apellidos, código de estudiante y descripción de carrera, junto con párrafo de resumen indicando principales conocimientos técnicos y habilidades que puede aportar en el equipo.

Solution Profile

Esta sección incluye dos secciones internas. La primera parte, Antecedentes y Problemática, consta del enunciado de problema, y una descripción de los puntos más importante que debe resolver la solución propuesta, así como objetivos y restricciones que delimiten el alcance del proyecto. La segunda parte, Lean UX

¹ Un **startup** es una pequeña empresa de reciente creación, con alto potencial innovador y tecnológico, donde su modelo es escalable y su crecimiento puede ser exponencial. En su traducción del inglés, el término start-up significa “puesta en marcha”. Y, efectivamente, podemos definirlo como el periodo inicial de una empresa, el comienzo o arranque de un nuevo negocio.

Process, es resultado de la ejecución del Lean UX Process sobre el dominio del problema.

Antecedentes y Problemática

Aquí se incluye una aproximación preliminar a la descripción de los antecedentes y la descripción de la problemática. Para la elaboración de esta descripción, el equipo debe aplicar previamente la técnica de The 5 'W's y 2 'H's - Who, What, Where, When, Why, How & How Much.

Lean UX Process

Aquí se aplica Lean UX Process y abarca la visión del modelo de negocio que será soportado por el producto de software, incluyendo Problem Statements (incluyendo aspectos como domain, customer segments, pain points, gap, visión/strategy, e initial segment), Assumptions e Hypothesis Statements según Lean UX Process. Finalizando esta sección se incluye el Lean UX Canvas.

Segmentos objetivo

Esta sección incluye la descripción de los segmentos asociados al dominio del problema, incluyendo características demográficas e información estadística de sustento.

Requirements Elicitation & Analysis

Se incluye el proceso de Needfinding junto con análisis de la competencia. Las entrevistas se registrarán en video y se editarán para construir el video de evidencia de entrevistas. El análisis de dichas entrevistas servirá de base para la identificación de necesidades y la construcción de los User Persona para cada segmento objetivo, así como la construcción del User Task Matrix, los User Journey Map para los User Persona identificados, así como los Empathy Maps, Big Picture EventStorming y Ubiquitous Language.

Competidores

En esta sección se realiza la identificación y descripción de los principales competidores directos (3 como mínimo) con modelos de negocio basados en productos digitales similares, o en su defecto competidores indirectos con ofertas parcialmente similares.

Análisis competitivo

En esta sección tiene como objetivo que su startup conozca mejor a sus competidores, en contraste con la idea inicial que pudiera tener sobre ellos. Se debe desarrollar el siguiente Landscape:

Competitive Analysis Landscape	
¿Por qué llevar a cabo este análisis?	Escriba en el recuadro la pregunta que busca responder o el objetivo de este análisis.

(En la cabecera colocar por cada competidor nombre y logo)		Su startup	Competidor 1	Competidor 2	Competidor 3
Perfil	Overview				
	Ventaja competitiva ¿Qué valor ofrece a los clientes?				
Perfil de Marketing	Mercado objetivo				
	Estrategias de marketing				
Perfil de Producto	Productos & Servicios				
	Precios & Costos				
	Canales de distribución (Web y/o Móvil)				
Análisis SWOT	Realice esto para su startup y sus competidores. Sus fortalezas deberían apoyar sus oportunidades y contribuir a lo que ustedes definen como su posible ventaja competitiva.				
	Fortalezas				
	Debilidades				
	Oportunidades				
	Amenazas				

Para cada uno de ellos debe identificarse fortalezas y debilidades, así como las oportunidades y amenazas asociadas.

Estrategias y tácticas frente a competidores.

Se debe incluir las estrategias y tácticas preliminares que aplicará su startup para afrontar las fortalezas y aprovechar las debilidades, así como el contexto de oportunidades y amenazas en relación a la competencia.

Entrevistas

En esta sección se aborda la investigación tomando como base la recolección de información en base a entrevistas a representantes de los segmentos objetivo.

Diseño de entrevistas

Esta sección incluye la relación de preguntas principales y complementarias para entrevistas, dirigidas a cada uno de los segmentos. Es importante considerar que debe aplicarse buenas prácticas para diseño de entrevistas. También debe considerar qué tipo de información principal y complementaria necesita recolectar para construir los arquetipos (características demográficas como género, edad, distrito de residencia, estado civil, familia, ocupación, al igual que otras características como personalidad, habilidades, marcas e influencias, dispositivos de preferencia, canales digitales de interacción, objetivos, frustraciones, biografía o background).

Registro de entrevistas

Para cada segmento se requiere de 3 a 5 entrevistas. Para cada una de las entrevistas se debe indicar la información de nombres, apellidos, edad, distrito, un screenshot de un cuadro de video y el URL del video subido en Microsoft Stream/Clipchamp (es un solo video editado para todas las entrevistas) incluyendo el timing donde inicia la entrevista y su duración. La entrevista debe ser registrada en video, que sirve de evidencia de entrevistas. Para cada entrevista debe redactarse en este informe un resumen, que explique de forma descriptiva las respuestas del entrevistado a las preguntas realizadas. Todas las características objetivas y subjetivas, incluyendo aspectos como personalidad, marcas e influencias, tecnología, canales de interacción, browser, dispositivos, etc. Deben estar incluidas como parte de los resúmenes para cada entrevista. Debe ser evidente que cada característica de los arquetipos que se construirán en base a esta información provienen de datos recolectados. Ver otras indicaciones importantes en el *Anexo C. Indicaciones para secciones que incluyen Videos*.

Análisis de entrevistas

En esta sección se debe realizar un análisis por cada segmento objetivo, identificando con sustento estadístico (porcentajes) todas las características objetivas y subjetivas que representan los aspectos más comunes de cada segmento y que son necesarios para la construcción de los arquetipos. La fuente de información para este análisis proviene de las entrevistas registradas. Debe evidenciarse que cada característica tiene relación con las entrevistas registradas y los resúmenes realizados para las mismas.

Needfinding.

En esta sección el equipo explica y presenta los artefactos resultantes del proceso de análisis de la información recolectada. Aquí se incluye secciones internas para User Personas, User Task Matrix, User Journey Maps, Empathy Mapping y As-Is Scenario Mapping.

User Personas

En esta sección se incluye la elaboración de las fichas de User Persona. La sección inicia con una introducción explicando la relación entre los artefactos a presentar y las principales características que se están tomando en cuenta del análisis de entrevistas y de la competencia. Se elabora una ficha de User Persona por cada segmento objetivo. Considere las mejores prácticas y todos los ítems necesarios para especificar un arquetipo. Utilice la herramienta indicada para este tipo de artefacto.

User Task Matrix

En esta sección se presenta el User Task Matrix, que concentra las tareas que los User Persona (que representan a cada segmento) realizan para cumplir sus objetivos. No confundir tareas (tasks) con opciones o características de software, pues las tareas deben ser realizadas por los segmentos independientemente de la existencia de su solución de software. Esta sección inicia con una introducción donde se establece los segmentos que se están considerando. El cuadro debe incluir como columna cada User Persona y para cada una como sub-columnas, la Frecuencia y la Importancia de cada tarea (task). Como filas se colocan las tareas identificadas. Luego del cuadro se realiza una explicación resaltando las tareas con mayor frecuencia e importancia, principales diferencias y coincidencias entre lo realizado por los User Personas.

User Journey Mapping

En esta sección se elabora los User Journey Maps (uno por cada User Persona). La sección inicia con una introducción que resume el end-to-end journey que se pretende ilustrar. Debe incluirse capturas de imagen de los diagramas elaborados en la herramienta indicada. En este caso se elabora las versiones As-Is de los User Journey Maps, es decir los journey de cada segmento representado para la situación actual, sin que exista su solución. Cada User Journey Map debe vincularse con el User Persona correspondiente (cuya ficha de User Persona también debe haberse elaborado en la misma herramienta indicada).

Empathy Mapping

En esta sección, el equipo resume el proceso de elaboración y presenta capturas de los Empathy Maps realizados en la herramienta indicada, para cada uno de los User Personas. El proceso de elaboración incluye la preparación, colocar al centro el User Persona. Colocar en la sección correspondiente en la herramienta cada observación de los miembros del equipo sobre el User Persona, buscando responder las preguntas ¿Con quién estamos empatizando? ¿Qué necesita hacer? ¿Qué está diciendo? ¿Qué está viendo? ¿Qué está haciendo? ¿Qué está escuchando? ¿Cómo se siente y qué piensa? Identificar Pains y Gains en base a las preguntas ¿Qué le

preocupa? Y ¿Qué puede ayudar a resolver sus problemas? ¿Qué puede convencerlo de que somos la alternativa correcta? ¿Qué dice?

Big Picture Event Storming

En esta sección el equipo introduce, resume el proceso realizado por el equipo y presenta capturas y explicaciones de las etapas del Big Picture Event Storming. En una sesión colaborativa, el equipo se enfoca en entender el dominio del negocio en general, plasmando los eventos significativos y sus relaciones. Es una primera aproximación visual de alto nivel que explora el landscape del negocio, identificando procesos clave, exponiendo potenciales problemas u oportunidades. En <https://bit.ly/bpes-guide> encontrará un Step-by-Step Guide para realizar el proceso.

Ubiquitous Language

En esta sección el equipo redacta un glosario de términos y conceptos con definiciones utilizadas en el business domain, sin ambigüedad, relacionados al área de especialidad o sector en el que se establecen el problema y la solución. Mantener un glosario de este tipo completo y actualizado, permite que se comuniquen claramente todos los miembros y stakeholders en el equipo. Solo debe incluirse términos del dominio, no términos técnicos del área de ingeniería de software. Los términos deben estar en inglés (puede incluirse adicionalmente el término equivalente en español entre paréntesis). La definición correspondiente al término puede estar en español. Eric Evans habla sobre *Ubiquitous Language* en su libro *Domain-Driven Design: Tackling Complexity in the Heart of Software*².

Requirements Specification

Esta sección permite que el equipo realice en base al análisis de la información obtenida en las investigaciones, la especificación de los requisitos de los productos digitales. La sección inicia con una introducción e incluye secciones internas para los User Stories, Impact Map y Product Backlog.

User Stories

Requisitos definidos junto con el conjunto de User Stories y Epics para los requisitos identificados. Los User Stories incluyen Acceptance Criteria. En esta sección el equipo redacta una introducción y presenta un cuadro con la estructura especificada a continuación. Solo se elabora un cuadro para todo el conjunto de Epics/Stories. Debe dedicar una línea del cuadro a cada Epic / User Story.

Epic / Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)

Es importante recordar que una User Story tiene varios criterios de aceptación. Los criterios de aceptación deben redactarse en tiempo presente, tercera persona, no hacer referencia a detalles de interfaz de usuario y deben ser comprobables. La

² En el libro Eric Evans establece “Ubiquitous Language is modeled within a Limited context, where the terms and concepts of the business domain are identified, and there should be no ambiguity”.

estructura de criterios de aceptación debe seguir la estructura de Gherkin (Given-When-Then). Las únicas excepciones podrían ser reglas de negocio o restricciones que no dependan de condiciones.

Debe también considerar User Stories para el sitio web estático (Landing Page), tomando como rol base *visitante* (o subconjuntos para cada segmento considerado cuando se requiera, como *visitante del segmento XXXX*) en la redacción de la descripción de la User Story. Recuerde que el sitio web estático tienen diversas secciones, con contenido dirigido a diversos segmentos, con características y comportamiento que permita su vínculo con la experiencia web.

Adicionalmente, debe considerarse Technical Stories para los features de productos digitales que no tienen interacción directa con los usuarios finales, como por ejemplo los RESTful APIs. En ese caso, utilice el rol *Developer* en la redacción de la descripción de la User Story. Considere como Acceptance Criteria los posibles escenarios de interacción de request/response, siguiendo la estructura de Gherkin.

Impact Mapping.

En esta sección el equipo explica y presenta capturas del Impact Mapping para el modelo de negocio digital, elaborado en la herramienta indicada. Para esto debe haber elaborado previamente en la herramienta las fichas para cada User Persona. La elaboración incluye la identificación de los *Business Goals* (los business goals deben cumplir con los criterios SMART³. Por ejemplo “Alcanzar los 600 usuarios suscritos al plan A en el lapso de 8 meses.”). Debe considerar varios *Business Goals*. Debe incluir como *Actors/Personas* a los User Personas previamente identificados, según relaciones con los Business Goals, buscando responder la pregunta ¿Quiénes me ayudarán a lograr la meta? La columna *Impact* debe incluir los enunciados de cómo desea que los User Persona cambien o se comporten ¿Qué tendría él/ella que hacer para ayudar a que se logre la meta? La columna *Deliverables* debe incluir los elementos que respondan la pregunta ¿Qué puedo hacer como negocio digital para provocar esos *Impacts*? La columna User Stories debe incluir la descripción de los User Stories (en el formato “Como... deseo... para...” que permitirán obtener los features que ayudarán a producir los *Deliverables* identificados.

Product Backlog.

Los User Stories deben incluir su estimación y priorización en el Product Backlog. Debe utilizar la herramienta indicada para el Product Backlog. Adicionalmente debe elaborar en este documento una tabla con la siguiente estructura.

# Orden	User Story Id	Título	Descripción	Story Points (1 / 2 / 3 / 5 / 8)
1	US01	AAA...	Como... deseo... para....	3

Adicionalmente debe incluir una captura y una referencia de URL del enlace público para el product backlog en la herramienta indicada. Recuerde que en el Product Backlog, el orden lo determina el valor para el negocio. Elaborar un product backlog colocando al inicio User Stories ligados a la seguridad o autenticación, por ejemplo,

³ Vea el artículo “Why are SMART Goals Necessary In Business?” en la sección de Referencias.

se considera incorrecto. Considere que los User Stories relacionados con el sitio web estático (Landing Page) requieren considerarse desde el primer sprint.

Strategic-Level Domain-Driven Design

En esta sección el equipo introduce y explica el proceso realizado para las decisiones de nivel estratégico aplicando Domain-Driven Design.

Bounded Contexts

En esta sección el equipo explica y evidencia el proceso para descomponer el sistema en subconjuntos con límites naturales o Bounded Contexts. Para ello debe aplicar las herramientas de EventStorming y Bounded Context Canvas.

EventStorming.

En esta sección el equipo explica y evidencia el proceso de Design-Level EventStorming, con el fin de plantear una primera aproximación revisada y mejorada al modelado de nivel general para el dominio del problema, buscando a partir de ahí identificar el mayor nivel de detalle posible. Es recomendable que el equipo organice la sesión de Design-Level EventStorming con una duración entre 1 – 2 horas, a fin de concentrar esfuerzos y no extender el proceso de forma innecesaria. La sección inicia con una introducción y explicación de las actividades realizadas en la sesión de EventStorming, e incluye capturas de lo elaborado en la herramienta indicada. Puede acceder a <https://bit.ly/dles-guide> para tener mayor referencia y guía sobre qué priorizar y cómo realizar el proceso.

Candidate Context Discovery

En esta sección el equipo, a partir del dominio modelado como EventStorm, explica y evidencia el proceso realizado para la sesión de Candidate Context Discovery, en la que se busca identificar los bounded contexts. Puede aplicar las técnicas de *start-with-value* (Identificar las partes core del dominio que tienen el mayor valor para el negocio), *start-with-simple* (Crear modelos simples, pero con propósito, descomponiendo el timeline en steps secuenciales), ó *look-for-pivotal-events* (Buscar eventos clave del negocio que indiquen cambios de estado entre diferentes partes del proceso de negocio). La sesión de Candidate Context Discovery no debería durar más de 2 horas. Utilice para el proceso la herramienta indicada. Complemente la explicación con capturas en imagen de los cambios progresivos del EventStorm.

Domain Message Flows Modeling.

En esta sección, el equipo explica y evidencia el proceso seguido para visualizar cómo deben colaborar los bounded contexts para resolver los casos que se presentan en el negocio para los usuarios del sistema. Para ello debe aplicar la técnica de visualización *Domain Storytelling*. Complemente la explicación con capturas en imágenes de los diagramas de Domain Storytelling elaborados.

Bounded Context Canvases

En esta sección el equipo diseña sus candidate bounded contexts, detallando los criterios de diseño. El equipo debe ir seleccionando cada bounded context, por orden de importancia, para elaborar su Bounded Context Canvas. La elaboración del

Bounded Context Canvas debe seguir un proceso iterativo con los pasos de *Context Overview Definition, Business Rules Distillation & Ubiquitous Language Capture, Capability Analysis, Capability Layering* (si aplica), *Dependencies Capture*, y *Design Critique*.

Al momento de la organización o refinamiento de bounded contexts es importante tomar en cuenta lo siguiente. En una plataforma SaaS (Software as a Service) orientada a negocios de servicio, es común encontrar los siguientes sub-dominios:

Subscriptions and Payment Management

Identity and Access Management

Profiles and Preferences Management

Service Design and Planning

Resource and Asset Management

Service Execution and Monitoring

Dashboard and Analytics

Loyalty and Engagement

Estos posibles sub-dominios pueden identificarse bajo otros nombres según la naturaleza o términos en el ubiquitous language del dominio en el que se enmarca la solución a realizar. Es posible que existan otros sub-dominios core o de soporte que se requiere considerar en el negocio objeto de estudio.

Context Mapping

En esta sección el equipo explica y evidencia el proceso de elaboración de un conjunto de contexts maps (visualizaciones de las relaciones estructurales entre bounded contexts). Para ello el equipo revisa información recolectada y la utiliza para producir los diseños candidatos. Se recomienda en el proceso incluir preguntas como: “¿qué pasaría si movemos este capability a otro bounded context?”, “¿qué pasaría si descomponemos este capability y movemos uno de los sub-capabilities a otro bounded context?”, “¿qué pasaría si partimos el bounded context en múltiples bounded contexts?”, “¿qué pasaría si tomamos este capability de estos 3 contexts y lo usamos para formar un nuevo context?”, “¿qué pasaría si duplicamos una funcionalidad para romper la dependencia?”, “¿qué pasaría si creamos un shared service para reducir la duplicación entre múltiples bounded contexts?”, “¿qué pasaría si aislamos los core capabilities y movemos los otros a un context aparte?”. Debe finalizar este proceso discutiendo cada alternativa de context mapping a fin de llegar a la mejor aproximación. Es importante que el equipo considere los patrones de relaciones entre Bounded Contexts establecidos en Domain-Driven Design, como *Anti-corruption Layer, Conformist, Customer/Supplier* ó *Shared Kernel*.

Software Architecture

En esta sección el equipo presenta y explica la representación, aplicando C4 Model y utilizando la herramienta indicada, de la Arquitectura de Software para la solución. Aquí se realiza una introducción y se incluye como secciones internas *Software Architecture Context Level Diagram* y *Software Architecture Container Level Diagrams*.

Software Architecture Context Level Diagram.

En esta sección el equipo realiza una introducción, presenta en imagen el context diagram, el cual debe mostrar el sistema como un recuadro en el centro, rodeado por sus usuarios y otros sistemas con los que interactúa. Utilice la herramienta indicada para la elaboración del diagrama. Se incluye en esta sección una explicación del diagrama.

Software Architecture Container Level Diagram

En esta sección, el equipo realiza una introducción, presenta y explica el Container Diagram. Dicho diagrama debe mostrar los elementos de alto nivel de la arquitectura de software y cómo se distribuyen las responsabilidades entre ellos. Aquí se debe mostrar también las principales decisiones de tecnología y cómo los containers se comunican entre sí.

Tactical-Level Software Design

En este capítulo el equipo explica y presenta su propuesta para la perspectiva táctica del diseño de la solución de software. Aquí se incluye una sección interna por cada bounded context.

Bounded Context: <Bounded Context Name>

En esta sección, el equipo presenta las clases identificadas y las detalla a manera de diccionario, explicando para cada una su nombre, propósito y la documentación de atributos y métodos considerados, junto con las relaciones entre ellas.

Domain Layer.

En esta capa el equipo explica por medio de qué clases representará el core de la aplicación y las reglas de negocio que pertenecen al dominio para el bounded context. Aquí el equipo presenta clases de categorías como *Entities*, *Value Objects*, *Aggregates*, *Factories*, *Domain Services*, o abstracciones representadas por interfaces como en el caso de *Repositories*.

Interface Layer.

En esta sección el equipo introduce, presenta y explica las clases que forman parte de Interface/Presentation Layer, como clases del tipo *Controllers* o *Consumers*.

Application Layer

En esta sección el equipo explica a través de qué clases se maneja los flujos de procesos del negocio. En esta sección debe evidenciarse que se considera los capabilities de la aplicación en relación al bounded context. Aquí debe considerarse clases del tipo *Command Handlers* e *Event Handlers*.

Infrastructure Layer

En esta capa el equipo presenta aquellas clases que acceden a servicios externos como *databases*, *messaging systems* o *email services*. Es en esta capa que se ubica la implementación de *Repositories* para las interfaces definidas en Domain Layer. Algo similar ocurre con interfaces definidas para *MessageBrokers*.

Bounded Context Software Architecture Component Level Diagrams.

En esta sección, el equipo explica y presenta los Component Diagrams de C4 Model para cada uno de los *Containers* considerados para el bounded context. En estos diagramas el equipo busca reflejar la descomposición de cada Container para identificar los bloques estructurales principales y sus interacciones. Un Component Diagram debe mostrar cómo un container está conformado por components, qué son cada uno de dichos components, sus responsabilidades y los detalles de implementación/tecnología. Utilice la herramienta indicada para la elaboración del diagrama.

Bounded Context Software Architecture Code Level Diagrams

En esta sección, el equipo presenta y explica los diagramas que presentan un mayor detalle sobre la implementación de componentes en el bounded context. Aquí se incluye como secciones internas *Bounded Context Domain Layer Class Diagrams* y *Bounded Context Database Diagram*.

Bounded Context Domain Layer Class Diagrams

En esta sección el equipo presenta el Class Diagram de UML para las clases del Domain Layer en el bounded context. El nivel de detalle debe incluir además de las clases, interfaces, enumeraciones y sus relaciones, los miembros para cada clase, incluyendo atributos, métodos y el scope en cada caso (private, public, protected). Las relaciones deben incluir la calificación con nombres, la dirección (cuando aplica) y la multiplicidad. Utilice para la elaboración del diagrama la herramienta indicada.

Bounded Context Database Diagram.

En esta sección el equipo presenta y explica el Database Diagram que incluye los objetos de base de datos que permitirán la persistencia de información para los objetos del bounded context. Para el caso de un almacenamiento en base de datos relacional, aquí debe especificarse tablas, columnas, constraints (por ejemplo, primary, foreign key) y evidenciarse las relaciones entre tablas. Utilice para la elaboración del diagrama la herramienta indicada.

Solution UI/UX Design

En esta sección se abarca el planteamiento de la propuesta de UX/UI Design para la experiencia de usuario a través de todos los productos digitales con los que estos interactúan. Para ello se tomará como base el conjunto de User Stories identificados así como el Impact Map.

Style Guidelines

En esta sección, el equipo sienta las bases para contar con un repositorio central y organizado de uso común para todo el equipo, que incluye assets, fonts, etc. Esto con el fin de mantener una presentación consistente y enfocada. Se incluye secciones para General Style Guidelines, Web Style Guidelines y Mobile Style Guidelines.

General Style Guidelines

Aquí se explica las decisiones y referencias visuales sobre conceptos generales básicos como Branding, Typography, Colors y Spacing, así como las dimensiones a adoptar para el tono de comunicación y lenguaje aplicado (Divertido/Serio, Formal/Casual, Respetuoso/Irreverente, Entusiasta/Sereno). Puede tomarse como referencia un Design System existente, sobre el cual se puede realizar adaptaciones. Esta sección debe incluir el sustento de principios y elementos de diseño considerados para las decisiones.

Web, Mobile and IoT Style Guide.

En esta sección se explica e ilustra las decisiones sobre los estándares visuales y de interacción para responsive web interfaces y las interfaces en mobile application interfaces y IoT application user interface.

Information Architecture

En esta sección el equipo plantea las decisiones y sustento que dirigen la manera como se organizará el contenido en las experiencias web y móvil, incluyendo el Landing Page y las Aplicaciones. Dichas propuestas deben estar orientadas a que los visitantes y usuarios se adapten con facilidad a la funcionalidad de cada producto y puedan encontrar todo aquello que necesiten sin esfuerzo. Se incluyen las decisiones sobre los Organization Systems, Labeling Systems, Navigation Systems y Searching Systems.

Organization Systems.

En esta sección el equipo explica en qué grupos de información aplicará cuáles sistemas de organización. Aquí se incluye la explicación de en qué casos se aplicará la organización visual del contenido: de forma jerárquica (visual hierarchy), organización secuencial (step-by-step to accomplish) o matricial. Por otro lado, también se debe explicar en qué casos se utilizará qué esquemas de categorización de contenido: alfabético, cronológico, por tópicos, según audiencia (grupos de usuarios).

Labeling Systems.

Aquí el equipo explica de qué maneras se representarán los datos, considerando simplicidad y buscando evitar la confusión para los visitantes y usuarios. En esta sección se especifica las etiquetas (con el mínimo número de palabras) a utilizar para representar los conjuntos de información y las asociaciones⁴ entre las mismas.

SEO Tags and Meta Tags

En esta sección se debe incluir los SEO Tags y Meta Tags junto con los valores que asignará en las principales páginas de la experiencia tanto a nivel del sitio web estático (Landing Page) como Web Application. Se debe incluir Title, los Meta Tags Description, Keywords, Author como mínimo. Del mismo modo, para las aplicaciones móviles u otros productos digitales que se exponen vía un app store, debe incluir los

⁴ Por ejemplo, la etiqueta 'Contacto' en un botón en el encabezado de una página sirve para asociar en la mente del visitante que encontrará en otro lugar información de contacto como número de teléfono, email y cuentas de redes sociales, sin necesidad de que todo esté aglomerado en un mismo lugar.

ASO (App Store Optimization) elements como App Title, App keywords, App subtitle, App description.

Navigation Systems

Aquí el equipo explica cuáles serán las acciones y técnicas que guiarán a los usuarios a través del Landing Page y las aplicaciones, permitiéndoles cumplir sus metas e interactuar de forma satisfactoria con el producto. Aquí se debe incluir de qué maneras los usuarios irán recorriendo el contenido.

Searching Systems

En esta sección el equipo explica qué medios de ayuda se brindará al usuario para la búsqueda de datos dentro del producto digital. Dichas decisiones sobre los sistemas de búsqueda tratan de evitar que los usuarios se sientan perdidos entre el volumen de información. Aquí se deben especificar qué opciones de búsqueda ofrecerán las aplicaciones, con qué filtros contará el usuario en cada caso y cómo lucirán los datos después de la búsqueda.

Landing Page UI Design

En esta sección el equipo elabora la propuesta de UI para el Landing Page. La sección inicia con una introducción en la que el equipo explica cómo traduce las decisiones de diseño y arquitectura de información.

Landing Page Wireframe

Esta sección incluye una sección interna donde se presenta y explica los Wireframes del Landing Page para Desktop Web Browser y Mobile Web Browser. En la propuesta y la explicación debe evidenciarse la aplicación de los principios, elementos de diseño, diseño inclusivo y arquitectura de información.

Landing Page Mock-up

Esta sección presenta y explica los Mock-ups del Landing Page, tanto en su versión para Desktop Web Browser como Mobile Web Browser. En la propuesta y la explicación debe evidenciarse la aplicación de los principios, elementos de diseño, diseño inclusivo y arquitectura de información, así como el Design System establecido para los productos digitales.

Applications UX/UI Design

Esta sección incluye secciones internas donde se presenta y explica la propuesta visual y de interacción para las aplicaciones que constituyen la experiencia de usuario con los productos digitales.

Applications Wireframes

Esta sección incluye una sección interna donde se presenta y explica los Wireframes de las aplicaciones móviles. En la propuesta y la explicación debe evidenciarse la aplicación de los principios, elementos de diseño, diseño inclusivo y arquitectura de información. Utilizar para los wireframes las herramientas indicadas.

Applications Wireflow Diagrams

Esta sección presenta la propuesta de Wireflows. Debe considerarse un Wireflow para cada User goal, considerando los User Persona para cada aplicación que forma parte del alcance. Es recomendable que el equipo elabore previamente los correspondientes Task Flows, para establecer un consenso sobre las rutas típicas de steps para cada User goal. Es importante recordar que la forma como se refleja un cambio en una pantalla (Wireframe) como resultado de la interacción en un flujo es agregar un paso con un Wireframe con la representación del nuevo estado. Utilizar para los Wireflows las herramientas indicadas. Cada Wireflow diagram requiere que se redacte el User goal y se complemente con una explicación del flujo especificado.

Applications Mock-ups

Esta sección presenta y explica los Mock-ups de las aplicaciones. En la propuesta y la explicación debe evidenciarse la aplicación de los principios, elementos de diseño, diseño inclusivo y arquitectura de información, así como el Design System establecido para los productos digitales. Utilizar para los mock-ups las herramientas indicadas.

Applications User Flow Diagrams.

Esta sección presenta la propuesta de User Flows. Debe considerarse un User Flow para cada *User goal*, considerando los User Persona para cada aplicación que forma parte del alcance. Estos User Flows deben ser consistentes con los Wireflows de los cuales se derivan. Debe recordarse que en el User Flow se incluyen los Mock-ups de las vistas o pantallas de las aplicaciones, junto con los flujos que constituyen la ruta esperada (happy path) y las rutas alternativas (unhappy paths). Utilizar para los User Flows las herramientas indicadas. Cada User Flow diagram requiere que se redacte el User goal y se complemente con una explicación de los flujos y condiciones especificados.

Applications Prototyping.

Esta sección incluye Prototipos de UI para Desktop y Mobile Web Browser con simulación de interacción y navegación, acorde con la propuesta de paths de User Flow Diagrams. Esta sección inicia con una introducción en la que se explica los principales criterios para las decisiones de interacción. Es importante evidenciar la relación con las decisiones de arquitectura de información, en particular sobre el sistema de navegación y los tipos de interacciones seleccionadas. Para cada caso debe incluirse 1 screenshot de video y un enlace a un video subido a Microsoft Stream/Clipchamp para cada aplicación, en el que se demuestre y explique los principales flujos de interacción que cubren los prototipos.

Product Implementation, Validation & Deployment

En esta sección el equipo explica y evidencia el proceso de implementar, comprobar, desplegar y validar la solución compuesta en este caso por los productos digitales que forman parte del alcance. El Landing Page permite presentar el modelo de negocio y las aplicaciones web. Los procesos del negocio digital que dirigen la operación del negocio, tanto procesos core del negocio como procesos de soporte (por ejemplo Authentication & Authorization, Subscriptions, entre otros) están distribuidos entre los productos digitales que forman parte del alcance como por ejemplo RESTful Web Services, Native Mobile Applications, Web Applications, Edge Web Services, Embedded Applications u otros. Este capítulo abarca secciones para la organización del proceso de trabajo en Sprints, la descripción y prácticas asociadas a Software Configuration Management, el Video About-The-Product y las evidencias de Implementation, Testing, Despliegue y Validaciones para cada uno de los productos que forman parte de la solución, en términos del producto en sí y la colaboración por Sprint.

Software Configuration Management

En esta sección el equipo establece las decisiones y convenciones que permitirán mantener la consistencia durante el ciclo de vida. Se incluyen secciones internas para Source Code Management, Development Environment Configuration y Deployment Configuration.

Software Development Environment Configuration

En esta sección el equipo especifica, describe e indica los nombres de productos, el propósito de uso en el proyecto, la ruta de referencia (para software basado en modelos SaaS) o ruta de descarga (para productos que se ejecutan en el computador del miembro del equipo) de cada uno de los productos de software que deben utilizar los miembros del equipo para colaborar en el ciclo de vida de los productos digitales que forman la solución con IoT, considerando todos los tipos de actividades como Project Management, Requirements Management, Product UX/UI Design, Software Development, Software Testing, Software Deployment, Software Documentation, respetando las restricciones indicadas sobre productos de software y herramientas que se pueden utilizar.

Source Code Management

En esta sección el equipo establece los medios y esquema de organización que aplicará para el seguimiento de modificaciones. Para ello utilizará GitHub como plataforma y sistema de control de versiones. Debe incluirse el URL del repositorio de GitHub para cada producto: Landing Page, Web Services, Frontend Web Applications. Tomar en cuenta que en el caso de Web Services, se incluye en el repositorio el proyecto y los archivos de pruebas, tanto unitarias como de integración/aceptación.

En esta sección debe también explicarse de qué forma implementará GitFlow (Ver artículo “A successful Git branching model” de Vincent Driessen en la sección de Referencias) como Workflow de control de versiones, es decir qué branches (ramas) creará además de main branch (rama principal), por ejemplo, develop branch. Para GitFlow cada Feature requiere su propio branch, por ello debe especificar qué

convenciones se aplicará para nombrar los feature branches. Igualmente debe incluir las convenciones para Release branches y Hotfix branches. Aplique semantic versioning para nombrar sus Releases (Vea “Semantic Versioning 2.0.0” en la sección de Referencias).

Aplique Conventional Commits para los textos de mensajes en sus commits (Vea “Conventional Commits” en la sección de Referencias).

Source Code Style Guide & Coding Conventions

Aquí el equipo explica e indica las referencias que adoptará para nombrar elementos y programar en los lenguajes que se utilizan en la solución (en este caso HTML, CSS, JavaScript, TypeScript, Java, C#, Kotlin, Swift, C++, Python u otros según los constraints del proyecto; así como Gherkin para los archivos .feature). Para todos los lenguajes debe aplicar la nomenclatura en inglés. Adicionalmente, adopte convenciones estándares para coding (Vea por ejemplo “HTML Style Guide and Coding Conventions”, “Google HTML/CSS Style Guide”, “Gherkin Conventions for Readable Specifications”, “Angular coding style guide”, “Google Java Style Guide”, “Google TypeScript Style Guide” y “Spring Boot Features” en la sección de Referencias).

Software Deployment Configuration

En esta sección el equipo especifica la configuración del despliegue de la solución, incluyendo los pasos necesarios para que, a partir de los repositorios de código fuente, se pueda lograr el despliegue o publicación satisfactorio de cada uno de los productos digitales en la solución (Landing Page, Web Services, Web Applications, Mobile Applications, Embedded Applications u otros productos incluidos). Adicionalmente a la explicación, el equipo incluye aquí el Deployment Diagram de C4 Model.

Video About-the-Product

En esta sección el equipo redacta una introducción y resumen del contenido incluido en el Video About-the-Product, el cual tiene como público objetivo los visitantes al Landing Page, quienes desean conocer sobre el modelo de negocio y las características principales de los productos de software, al igual que los usuarios de las Aplicaciones, quienes desean realizar tareas relacionadas con los procesos soportados por la solución. El tono que utilice en la comunicación debe ser consistente con el tono adoptado para el producto y debe incluirse al menos un testimonio positivo de un usuario que haya participado en las entrevistas de validación. Debe incluirse también en esta sección un screenshot del Video, el URL de la versión publicada en Microsoft Stream/Clipchamp (y además, el URL de la versión publicada en YouTube utilizada para incrustarse en el Landing Page), así como el timing (duración) del mismo. Ver otras indicaciones importantes en el *Anexo C. Indicaciones para secciones que incluyen Videos*.

Landing Page, Services & Applications Implementation.

En esta sección se explica y evidencia el proceso de implementación, pruebas, documentación y despliegue del Landing Page, Web Services, Web Applications, Mobile Applications y Embedded Applications. En esta sección se incluye, una vez

que se cuenta con el Product Backlog, una sección interna cada Sprint (Sprint 1, Sprint 2, etc.).

Sprint n

En esta sección se registra y explica el avance en términos de producto y trabajo colaborativo para el Sprint n. Incluye como secciones internas: *Sprint Planning n*, *Sprint Backlog n*, *Development Evidence for Sprint Review*, *Execution Evidence for Sprint Review*, *Services Documentation Evidence for Sprint Review*, junto con *Team Collaboration Insights during Sprint*.

Sprint Planning n

En esta sección se especifica los aspectos principales del Sprint Planning Meeting. Se inicia la sección con una introducción y a continuación se coloca el cuadro de resumen del sprint planning meeting. La estructura a utilizar se presenta a continuación.

Sprint #	Sprint n
Sprint Planning Background	
Date	YYYY-MM-DD
Time	HH:MM AM/PM
Location	(Descripción de la ubicación de la reunión, física o virtual)
Prepared By	Jiménez Rosas, Arturo Eduardo
Attendees (to planning meeting)	Jiménez Rosas, Arturo Eduardo / Rodríguez Peña, Jorge Andrés / ...
Sprint n – 1 Review Summary	(Resumen del Sprint anterior, en términos de resultados alcanzados a nivel de productos de software, opiniones de miembros y feedback de product owner.)
Sprint n – 1 Retrospective Summary	(Resumen del Sprint anterior, en términos de opiniones de miembros del equipo sobre aciertos u oportunidades de mejora en su forma de trabajo)
Sprint Goal & User Stories	
Sprint n Goal	(Definir el Goal del Sprint n y la métrica de cumplimiento.)
Sprint n Velocity	(Definir el Velocity establecido para el Sprint n, es decir cuántos Story Points puede aceptar el equipo para este Sprint n.)
Sum of Story Points	(Colocar la suma de los Story Points para los User Stories que se están incluyendo en este Sprint n.)

Es muy importante que el equipo dedique atención a la identificación del Sprint Goal. Según el Scrum Guide “El Sprint Goal es el objetivo individual del Sprint. Es un compromiso para los Developers, flexible en términos del trabajo exacto que se

requiere para alcanzarlo. El Sprint Goal también crea coherencia y enfoque, buscando que los miembros del Scrum Team trabajen juntos en vez de ir en pos de iniciativas individuales.”

Para identificar el Sprint Goal, es recomendable enfocarse en el negocio (business) o la perspectiva de los usuarios (user-focused), como por ejemplo entregar un nuevo feature o feature set.

Escribir objetivos SMART (Specific, Measurable, Attainable, Relevant, Time-bound) puede ser de mucha utilidad para que el equipo articule el Sprint Goal.

Scrum.org ofrece un template para redactar Sprint Goals. La estructura es:

*Our focus is on <Outcome>
We believe it delivers <Impact> to <Customer(s)>
This will be confirmed when <Event happens>*

Por ejemplo, tomando como ejemplo un solo feature: “**Our focus** is on sending a basic email that contains a link to a spreadsheet. **We believe it delivers** confidence in the product to our organization. **This will be confirmed when** we have an email in an inbox”.

Es importante que el equipo se enfoque en identificar *Outcome*, *Impact*, *Customer(s)*, así como *Event*. El solo hecho de aplicar el template no garantiza un buen Sprint Goal. El siguiente ejemplo es un uso **incorrecto** del template: “**Our focus is on** having SAP integrated into the Corporate system. **We believe it delivers** satisfaction and closure for our project manager. **This will be confirmed when** Epic SAP-123 is closed in Jira.” Aquí el Goal se está centrando en complacer a alguien del equipo. Una mejor aproximación a un Sprint Goal sería: “**Our focus is on** having SAP integrated into the Corporate system. **We believe it delivers** improved features functionality and increasing productivity to Department X. **This will be confirmed when** the colleagues in Department X are using the improved features and can see the benefits.”

Es recomendable que la redacción sea específica en términos de qué features se compromete a lograr y cómo benefician a los segmentos objetivo, sin detallar cómo. Algunos ejemplos de feature-sets con un enfoque de alto nivel:

“A customer can place an order from a single-product catalog”.
“Show a multi-product catalog”.
“Show top-selling products”.
“Make product catalog manageable for the sales department operator”

Es muy importante que el equipo establezca en conjunto el Sprint Goal, pues de esa forma todo el equipo puede trabajar en identificar qué epics y stories deberían considerarse en principio en la iteración, en base a su contribución para ese Goal en particular.

A continuación dos ejemplos de Sprint Goals. En cada ejemplo, se brinda un contexto previo para entender de mejor manera el propósito del Sprint Goal.

Example #1

Context

The CatchUp Product Team decided to add a new feature: To offer users the option to favorite news sources, for showing news from that sources in the new options “favorites”

Sprint Goal

“Our focus is on offering a more customized experience to the current CatchUp user community.

We believe it delivers rapid access to news from frequent news sources to users of the app.

This will be confirmed when users access news from news sources they previously marked as favorite ones in the app, with a Session Length of 10-minute or more.”

Example #2

Context

The ACME Learning Center Product Team decided to include plan information in the platform website, and allow students of the learning platform to cancel a previously requested enrollment. They also will implement a set of endpoints in order to allow frontend and mobile app developers the possibility of implementing student enrollment transaction features in their apps through the API.

Sprint Goal

“Our focus is on offering more detailed information to new visitors, providing an enhanced enrollment process to students, and incrementing new feature possibilities to the development team members.

We believe it delivers an easier plan selection to visitors, more flexibility on enrollment to students, and opportunities for implementing more enrollment-related features in the Learning Center ecosystem to development team

This will be confirmed when visitors can subscribe directly with and specific plan in no more than three steps, students can cancel previously requested enrollments without intervention of support staff, and developers implement new features related to enrollments using the implemented endpoints without intervention of backend development team.”

Aspect Leaders and Collaborators

En esta sección el equipo que incluye la elaboración de un artefacto Leadership-and-Collaboration Matrix (LACX), que indique por cada aspecto dentro del alcance del Sprint, quién es el líder y quién o quiénes son colaboradores en dicho aspecto, con el fin de brindar mayor claridad y efectividad en la comunicación al interior del equipo. La sección incluye una introducción donde se explica cuáles son los principales aspectos que se toma en cuenta en el Sprint. Dependiendo del Sprint un aspecto puede ser un subconjunto del alcance funcional de la solución (por ejemplo feature, bounded context, etc.). A continuación se incluye un cuadro con la siguiente estructura.

Team Member (Last Name, First Name)	GitHub Username	Aspect Name 1 Leader (L) / Collaborator (C)	Aspect Name 2 Leader (L) / Collaborator (C)	...	Aspect Name n Leader (L) / Collaborator (C)
Jiménez Rosas, Arturo Eduardo	ajimenezrosas	L	C	...	
Rodríguez Peña, Jorge Andrés	Japr91	C	C	...	L

La organización de líderes y colaboradores debe tener relación con la posterior selección de tasks en el Sprint.

Sprint Backlog n

Una sección de Sprint Backlog debe iniciar con una introducción que resuma el objetivo principal del Sprint y a continuación presente un screenshot del Board para el Sprint en la herramienta de control indicada (por ejemplo Trello), junto con el URL público del Board. A continuación, debe incluir una tabla donde se especifique los User Stories asignados al Sprint, junto con los Work-items/Tasks resultantes de la descomposición de los User Stories o Tasks adicionales que no dependen de un User Story en particular (por ejemplo, un task que debe realizarse para satisfacer un constraint general).

A continuación, la estructura de la tabla de control de estado para un Sprint.

Sprint #		Sprint n					
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation (Hours)	Assigned To	Status (To-do / In- Process / To- Review / Done)

Development Evidence for Sprint Review

En esta sección se explica y presenta los avances en implementación con relación a los productos de la solución según el alcance del Sprint: Landing Page, Web Applications, Web Services y otros. La sección inicia con una introducción que resume los principales avances en la implementación. Debe elaborarse una tabla que incluya para cada repositorio los commits relacionados con la implementación. La estructura requerida se presenta a continuación.

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
user/repositoryname	feature/loremipsum	14ca4e3	feat: consectetur adipiscing elit	Curabitur quis placerat nulla. Fusce malesuada faucibus quam, ut condimentum velit rutrum ut.	04/09/2021

Testing Suite Evidence for Sprint Review

En esta sección se explica y presenta el conjunto de Unit Tests, Integration Tests y Acceptance Tests automatizados, para Web Services relacionados con los User Stories especificados en el Sprint. En el caso de los tests de BDD debe elaborarse los

archivos .feature utilizando el lenguaje Gherkin y los archivos Steps en el lenguaje de programación. En esta sección se debe incluir la relación de tests diseñados. En el caso de los Unit Tests, debe indicarse con qué clases y comportamientos se relacionan. En el caso de los Integration Tests ó Acceptance Tests bajo el enfoque BDD, se incluye el código de los .feature Files, explicando con qué User Stories se relacionan. También debe incluirse la ruta del repositorio de control de versiones para los proyectos de Testing, junto con los id de commits relacionados con los avances en Testing para este Sprint. Debe elaborarse una tabla que incluya para cada repositorio los commits relacionados con testing. La estructura requerida se presenta a continuación.

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
user/repositoryname	feature/loremipsum	14ca4e3	test: consectetur adipiscing elit	Curabitur quis placerat nulla. Fusce malesuada faucibus quam, ut condimentum velit rutrum ut.	04/09/2021

Execution Evidence for Sprint Review

Esta sección inicia con un resumen que explique lo alcanzado en este Sprint y presenta screenshots de las principales vistas implementadas, junto con un enlace a un video que ilustre y explique la visualización y navegación logrados en este Sprint.

Services Documentation Evidence for Sprint Review.

En esta sección se incluye la relación de Endpoints documentados con OpenAPI, relacionados con el alcance del Sprint. La sección inicia con una introducción en la que se resume los logros alcanzados en relación con Documentación de Web Services para este Sprint. Debe elaborarse una tabla en la que se incluya, para cada Endpoint, la indicación de acciones implementadas, junto con los enlaces correspondientes a la documentación desplegada (o URL local en Sprints previos al despliegue de Web Services). Indicar las acciones soportadas incluyendo para cada acción el verbo http (get, post, put, delete, patch), sintaxis de llamada, especificación de posibles parámetros, así como ejemplo y explicación del response. Adicionalmente, debe incluirse y explicarse capturas en imágenes de la interacción, utilizando datos de muestra, con la documentación elaborada. Debe incluirse el URL del repositorio de Web Services, junto con los id de los commits relacionados con Documentación para este Sprint.

Software Deployment Evidence for Sprint Review.

En esta sección se resume los procesos realizados en relación con Deployment durante este Sprint. La sección inicia con una introducción explicando qué se ha realizado con respecto a despliegue durante este Sprint. Abarca actividades de creación de cuentas, configuración de recursos en cloud providers, configuración de proyectos de desarrollo para integración o automatización de labor de Deployment, entre otros. Se considera dentro del proceso de Deployment todos los productos digitales: Landing Page, Web Services, Aplicaciones y otros productos que formen

parte del alcance. Se debe adicionar capturas en imagen y explicaciones de los pasos realizados durante el Sprint.

Team Collaboration Insights for Sprint Review

En esta sección el equipo explica cómo se han desarrollado las actividades de implementación y se presenta capturas en imagen de los analíticos de colaboración y commits en GitHub, realizados por los miembros del equipo, así como la redacción de la interpretación de estos analíticos por parte del equipo. Todos los miembros del equipo deben tener participación en la implementación de cada uno de los productos según corresponda en el Sprint: Landing Page, Web Services y Aplicaciones.

Validation Interviews

En esta sección, el equipo registra y explica las actividades de entrevistas de validación durante el proyecto.

Se debe realizar entrevistas de validación en las que usuarios de los segmentos objetivo interactúen con el landing page y con las aplicaciones. Incluye secciones internas para Diseño de Entrevistas, Registro de Entrevistas, Evaluaciones según heurísticas. Para el proceso de validación debe aplicarse el formato de evaluación heurística indicado para el proyecto.

Diseño de Entrevistas.

En esta sección el equipo establece por cada segmento objetivo los elementos a incluir en la sesión de validación, incluyendo el Landing Page y las aplicaciones. Aquí se especifica también cuáles serán los user flows de las aplicaciones, que formarán parte del proceso de validación.

Registro de Entrevistas.

Para cada segmento se requiere de 3 a 5 entrevistas. Para cada una de las entrevistas se debe indicar la información de nombres, apellidos, edad, distrito, un screenshot de un cuadro de video y el URL del video subido en Microsoft Stream/Clipchamp incluyendo el timing donde inicia la entrevista y su duración. La entrevista debe ser registrada en video, que sirve de evidencia de entrevistas. Para cada entrevista debe redactarse en este informe un resumen, que explique de forma descriptiva las principales apreciaciones del entrevistado con respecto a las tareas asignadas. Ver otras indicaciones importantes en el *Anexo C. Indicaciones para secciones que incluyen Videos*.

Evaluaciones según heurísticas.

Esta sección contiene el proceso de evaluación de las sesiones de validación basado en heurísticas, considerando heurísticas de usabilidad, arquitectura de información e inclusive design de la experiencia propuesta. Para esto la sección debe contener la estructura del formato para evaluaciones de heurísticas indicado en el *Anexo D. Formato para Evaluación de User Experience según Heurísticas*.

Conclusiones

En esta sección tiene nivel equivalente a un capítulo. Aquí se incluye como secciones internas *Conclusiones y recomendaciones*, así como *Video About-The-Team*.

Conclusiones y recomendaciones

En esta sección el equipo enuncia las conclusiones sobre el trabajo, incluyendo los resultados a los que ha llegado en relación a los Problem Statements especificados, los assumptions realizados frente al comportamiento real de los segmentos, los Hypotheses Statements establecidos y los criterios de éxito especificados en el proceso de Lean UX, en contraste con los resultados obtenidos de las validaciones. Igualmente incluye recomendaciones sobre los siguientes pasos en relación a Roadmap de los productos digitales que forman parte del alcance del modelo de negocio digital.

Video About-The-Team

En esta sección el equipo redacta un resumen de los aspectos más relevantes del video About-The-Team, la pauta de secuencias de contenido (secciones con el timing de inicio de cada una, es decir hh:mm:ss de cada sección dentro del video) incluyendo además un cuadro de video representativo del mismo, junto con el URL de la versión publicada en Microsoft Stream/Clipchamp (y además, el URL de la versión publicada en YouTube utilizada para incrustarse en el Landing Page). Este video resume el proceso de trabajo realizado, incluyendo escenas con imágenes o video de sesiones de trabajo real del equipo, complementando con narración (voz en off) del proceso. Incluye además el testimonio ante cámara de cada participante describiendo actividades realizadas, logro de outcomes y desarrollo de competencias alcanzados. Ver otras indicaciones importantes en el *Anexo C. Indicaciones para secciones que incluyen Videos*.

Bibliografía

En esta sección el equipo especifica todas las referencias bibliográficas en formato APA 7ma edición (<https://normas-apa.org/>), utilizadas como base para el desarrollo del trabajo o referenciadas en secciones del informe.

Anexos

En esta sección, el equipo incluye como anexos tablas, documentos, gráficos, u otros elementos que por su extensión o grado de importancia ameriten aparecer en esta sección. Cada sección de anexo debe iniciar en una nueva página diferenciando el título con una letra mayúscula (Ejemplo: Anexo A, Anexo B, etc.)

Considere un Anexo con el nombre *Videos de Exposiciones*, donde se incluya de forma progresiva el título e hipervínculo al video de Exposición en Microsoft Stream/Clipchamp para cada entrega del proyecto. La relación de títulos y videos se va expandiendo con cada entrega.

El Equipo de Trabajo (Startup)

El equipo de desarrollo estará conformado por un grupo de estudiantes (el número de integrantes será indicado por el docente), entre quienes se distribuirá los roles y actividades a realizar como parte del proyecto. Es importante recalcar que

independientemente de la colaboración en los diversos aspectos relacionados al proyecto, todos los participantes deben colaborar en la elaboración de propuestas, implementación, pruebas y despliegue de las experiencias web, así como la implementación del sitio web estático para el landing page, evidenciando el desarrollo de las competencias objetivo de este curso.

Ética y responsabilidad

Se debe redactar los términos y condiciones de servicio, los cuales deben estar expuestos vía un enlace en el footer del Landing Page y el footer de las Aplicaciones. En la redacción debe evidenciarse responsabilidad ética y profesional, según los principios del código de ética de software engineering de ACM/IEEE y del CIP. Del mismo modo se debe evidenciar transparencia y responsabilidad profesional en el proceso del ciclo de vida del producto, evidenciando por escrito, vía capturas y en video según corresponda, el proceso de ingeniería de software que sigue el equipo, así como la aplicación de herramientas y servicios para el desarrollo colaborativo como el control de versiones de GitHub.

Internacionalización & Accesibilidad

Debe evidenciarse que el ciclo de vida de la solución y los productos elaborados están dirigidos por un enfoque inclusivo. En el caso de los productos, éstos deben incluir características de *Internationalization* bajo i18n (En el caso del Landing Page, Frontend Web Applications y Web Services) y *Accessibility* bajo a11y (en el caso del Landing Page y Frontend Web Applications). Considere como base los idiomas English (en_US), Latin American Spanish (es_419). Incluya en las experiencias web la configuración de ARIA attributes. Tome consideraciones adecuadas para internationalization y accessibility en los demás productos digitales de la solución. El idioma por defecto para los mensajes, interfaz de usuario e interfaz de documentación en todos los productos de la solución (Landing Page, Web Services, Aplicaciones) es **inglés**.

Tecnología

Para elaborar los *User Personas*, *Empathy Maps*, *Journey Maps* e *Impact Maps* se utilizará *UXPressia*.

Para la elaboración de Wireframes, Mock-ups y Prototypes, se utilizará *Figma*.

Para la elaboración de Wireflows y User Flows, se utilizará *LucidChart / Overflow*.

Para los diagramas de EventStorming se utilizará *LucidChart / Miro*.

Para la elaboración de diagramas de Software Architecture se utilizará *Structurizr* para C4 Model, *LucidChart* para UML y para Database Design se utilizará *LucidChart / Vertabelo*. En caso de aplicar Diagram-as-Code se puede utilizar *Structurizr DSL* para C4 Model y/o *PlantUML* para UML y C4 Model.

Para el desarrollo del Landing Page, se utilizará *HTML5*, *CSS3* y *JavaScript*.

Para el desarrollo de Web Applications, se hará uso de *Angular Framework*, o *Vue* utilizando *HTML5*, *CSS3* y *JavaScript* para aspectos estáticos de templates y *JavaScript* o *TypeScript* como lenguaje de programación. El lenguaje de diseño de Landing Page y Web Applications estará basado en *Material Design*. Como biblioteca de componentes de UI se utilizará *Angular Material*.

Para el desarrollo de Web Services, se realizará bajo *RESTful API* architectural style y se hará uso de *Spring Boot Framework*, *ASP.NET Core Framework*, o *Nest Framework*, utilizando *Java*, *C#* o *TypeScript* respectivamente como lenguaje de Programación.

Para la documentación de Web Services se utilizará *OpenAPI Specification* vía Swagger.

Para Edge Services, se utilizará Flask, Pewee ORM, SQLite, utilizando Python como lenguaje de Programación.

Para las Mobile Applications, se utilizará Kotlin en Android, Swift en iOS para Native Mobile Apps. Lo mismo aplica para OS de dispositivos IoT basados de Android o Apple, como Wearables, Smart TV, Smart Home Devices, o Car.

Si la naturaleza de la solución soporta el uso de Cross-Platform Mobile Apps, se utilizará Flutter con Dart, .NET MAUI con C#, Skip con Swift, o Kotlin Multiplatform con Kotlin.

Para las IoT Embedded Applications se utilizará C++, Python, o el lenguaje oficial recomendado por el fabricante del Embedded OS.

Para IoT Device Management, Cloud Computing y Edge Computing (para alguna característica complementaria al RESTful Edge API elaborado) puede utilizar las alternativas que ofrecen los cloud providers Microsoft, AWS, Google, u otras como Arduino, según la naturaleza de la solución.

Para el control de proyectos, se utilizará Pivotal Tracker / JetBrains YouTrack / Jira Software / Trello.

Para el almacenamiento y control de versiones de código se utilizará GIT gestionado desde GitHub aplicando GitFlow Workflow, Conventional Commits y Semantic Versioning.

Para el despliegue de aplicaciones móviles para su prueba en dispositivos físicos, se utilizará Firebase App Distribution.

Evaluación del Trabajo Final

El trabajo se ha dividido en 4 entregables.

TB1 – Sprint Review – Semana 4

Final Project Documentation Report

Final Project Keynote

Final Project Individual Member Performance Report (by Team Leader)

Archivo .zip con archivos complementarios según corresponda (videos, proyectos de software, documentos complementarios).

Video de Exposición

Consideraciones.

Incluir avance de Conclusiones, Bibliografía y Anexos.

Aspectos a incluir:

Carátula

Registro de Versiones del Informe

Project Report Collaboration Insights

Contenido

Student Outcome

Capítulo I: Introducción

Capítulo II: Requirements Elicitation & Analysis.

Capítulo III: Requirements Specification.

Capítulo IV: Solution Software Design.

Conclusiones

Bibliografía

Anexos.

TP1 – Stage Review – Semana 7

Final Project Documentation Report

Final Project Keynote

Final Project Individual Member Performance Report (by Team Leader)

Archivo .zip con archivos complementarios según corresponda (videos, proyectos de software, documentos complementarios).

Consideraciones.

Debe incluir versión corregida y mejorada de artefactos previamente presentados.

Debe incluir versión actualizada de Registro de Versiones del Informe, Project Report Collaboration Insights y Sección Student Outcome.

Debe incluir versión corregida y mejorada de artefactos previamente presentados.

A nivel de implementación debe estar implementada y desplegada la primera versión del Landing Page.

Debe estar desplegada la primera versión de Frontend Web Applications.

Debe incluir en el informe:

Capítulo V: Solution UI/UX Design

Capítulo VI: Product Implementation, Validation & Deployment.

6.1. Software Configuration Management.

6.1.1. Software Development Environment Configuration.

6.1.2. Source Code Management.

6.1.3. Source Code Style Guide & Conventions.

- 6.1.4. Software Deployment Configuration.
- 6.2. Landing Page, Services & Applications Implementation.
- 6.2.1. Sprint 1
 - 6.2.1.1. Sprint Planning 1.
 - 6.2.1.2. Aspect Leaders and Collaborators.
 - 6.2.1.3. Sprint Backlog 1.
 - 6.2.1.4. Development Evidence for Sprint Review.
 - 6.2.1.5. Testing Suite Evidence for Sprint Review.
 - 6.2.1.6. Execution Evidence for Sprint Review.
 - 6.2.1.7. Services Documentation Evidence for Sprint Review.
 - 6.2.1.8. Software Deployment Evidence for Sprint Review.
 - 6.2.1.9. Team Collaboration Insights during Sprint.

Conclusiones

Bibliografía

Anexos.

TB2 – Sprint Review – Semana 12

Final Project Documentation Report

Final Project Keynote

Final Project Individual Member Performance Report (by Team Leader)

Archivo .zip con archivos complementarios según corresponda (videos, proyectos de software, documentos complementarios).

Consideraciones

Debe incluir versión actualizada de Registro de Versiones del Informe, Project Report Collaboration Insights y Sección Student Outcome.

Debe incluir versión corregida y mejorada de artefactos previamente presentados.

Debe estar desplegada nueva versión del Landing Page.

Debe estar desplegada nueva versión de Web Applications.

Debe estar desplegada primera versión de otras Aplicaciones que forman parte del alcance.

Debe incluir en el informe:

- 6.2.2. Sprint 2
 - 6.2.2.1. Sprint Planning 2.
 - 6.2.2.2. Aspect Leaders and Collaborators.
 - 6.2.2.3. Sprint Backlog 2.
 - 6.2.2.4. Development Evidence for Sprint Review.
 - 6.2.2.5. Testing Suite Evidence for Sprint Review.
 - 6.2.2.6. Execution Evidence for Sprint Review.
 - 6.2.2.7. Services Documentation Evidence for Sprint Review.
 - 6.2.2.8. Software Deployment Evidence for Sprint Review.
 - 6.2.2.9. Team Collaboration Insights during Sprint.

6.3. Validation Interviews.

6.3.1. Diseño de Entrevistas.

6.3.2. Registro de Entrevistas.

6.3.3. Evaluaciones según heurísticas.

6.4. Video About-the-Product.

Conclusiones

Bibliografía

Anexos.

TF1 – Release Review – Semana 15

Final Project Final Documentation Report

Final Project Final Keynote

Final Project Final Individual Member Performance Report (by Team Leader)

Archivo .zip con archivos complementarios según corresponda (videos, proyectos de software, documentos complementarios).

Consideraciones

Debe incluir versión actualizada de Registro de Versiones del Informe, Project Report Collaboration Insights y Sección Student Outcome.

Debe incluir versión corregida y mejorada de artefactos previamente presentados.

Debe estar desplegada versión final de las Aplicaciones que forman parte del alcance.

Debe incluir en el informe versión final y evidencias de la conclusión del ciclo de vida y el proyecto con todos los Capítulos.

Debe incluir sección 6.2.3. Sprint 3.

Conclusiones

Bibliografía

Anexos.

Referencias

The Markdown Guide

<https://www.markdownguide.org/>

How to use PlantUML with Markdown

<https://gist.github.com/noamtamim/f11982b28602bd7e604c233fbe9d910f>

Structurizr - Embedding diagrams

<https://docs.structurizr.com/cloud/embed>

Seriously, what's your (startup's) problem?

<https://medium.com/@jakemendel/seriously-whats-your-startup-s-problem-b3a884c54ab4>

5W+2H - Técnica de análisis de problemas

<https://www.progressalean.com/5w2h-tecnica-de-analisis-de-problemas/>

Ubiquitous Language

<https://martinfowler.com/bliki/UbiquitousLanguage.html>

Ubiquitous Language: Unambiguously define the term and concepts of a business domain.

<https://openpracticelibrary.com/practice/ubiquitous-language/>

Lean UX – Chapter 3

<https://www.scribd.com/document/655516553/Leanux-Sampler>

Mike Cohn's Mountain Goat Software Blog – User Stories Articles

<https://www.mountangoatsoftware.com/blog/tag/user-stories>

User vs. Buyer Persona: Differences and free template

<https://uxpressia.com/blog/user-persona-vs-buyer-persona-difference>

How to create an impact map in 7 easy steps: A complete guide + examples

<https://uxpressia.com/blog/build-impact-map-4-easy-steps>

As-is Scenario Map: Build a better understanding of your users' current experience.

<https://www.ibm.com/design/thinking/page/toolkit/activity/as-is-scenario-map>

To-be Scenario Map: Draft a vision of your user's future experience to show how your ideas address their current needs.

<https://www.ibm.com/design/thinking/page/toolkit/activity/to-be-scenario-map>

Empathy Map: Build empathy for your users through a conversation informed by your team's observations.

<https://www.ibm.com/design/thinking/page/toolkit/activity/empathy-map>

Empathy Mapping: The First Step in Design Thinking

<https://www.nngroup.com/articles/empathy-mapping/>

How to build a customer empathy map (example + template)

<https://uxpressia.com/blog/empathy-map-free-template>

Figma Learn

<https://help.figma.com/hc/en-us/categories/23557013073047-Courses-tutorials-projects>

Acceptance Criteria in Scrum: Explanation, Examples, and Template

<https://dzone.com/articles/acceptance-criteria-in-software-explanation-examp>

A Beginner's Guide to finding User Needs

<https://jdittrich.github.io/userNeedResearchBook/>

Using a Requirements Traceability Matrix to improve project quality

<https://www.modernrequirements.com/blogs/using-a-requirements-traceability-matrix-to-improve-project-quality/>

Otiscole Customer Portfolio – What’s Cookin’

<http://otiscole.com>

A step-by-step guide to scenario mapping

<http://www.uxforthemasses.com/scenario-mapping/>

What are User Flows in User Experience (UX) Design?

<https://careerfoundry.com/en/blog/ux-design/what-are-user-flows/>

SEO and meta descriptions: Everything you need to know

<https://searchengineland.com/seo-meta-descriptions-everything-to-know-447910>

Design Systems 101

<https://www.nngroup.com/articles/design-systems-101/>

Front-End Style-Guides: Definition, Requirements, Component Checklist

<https://www.nngroup.com/articles/front-end-style-guides/>

The Four Dimensions of Tone of Voice

<https://www.nngroup.com/articles/tone-of-voice-dimensions/>

A successful Git branching model

<https://nvie.com/posts/a-successful-git-branching-model/>

Semantic Versioning 2.0.0

<https://semver.org/>

Conventional Commits

<https://www.conventionalcommits.org/>

HTML Style Guide and Coding Conventions

https://www.w3schools.com/html/html5_syntax.asp

Google HTML/CSS Style Guide

<https://google.github.io/styleguide/htmlcssguide.html>

Gherkin Conventions for Readable Specifications

<https://specflow.org/gherkin/gherkin-conventions-for-readable-specifications/>

Domain-Driven Architecture Diagrams

<https://medium.com/nick-tune-tech-strategy-blog/domain-driven-architecture-diagrams-139a75acb578>

Domain Storytelling and Requirements

<https://domainstorytelling.org/#dst-requirements>

Domain Driven Design: Tackling Complexity in the Heart of Software

<https://openpracticelibrary.com/perspective/domain-driven-design/>

Angular coding style guide

<https://angular.io/guide/styleguide>

Google Java Style Guide

<https://google.github.io/styleguide/javaguide.html>

Google TypeScript Style Guide

<https://google.github.io/styleguide/tsguide.html>

Spring Boot Features

<https://docs.spring.io/spring-boot/docs/current/reference/html/features.html>

C++ Core Guidelines

<https://isocpp.github.io/CppCoreGuidelines/>

MicroPython

<https://micropython.org/>

Python Style Guide

<https://www.python.org/doc/essays/styleguide/>

Python Style Guide for Arduino projects

<https://www.makerguides.com/python-style-guide-for-arduino-projects/>

Must-know HTML Tags for SEO and Digital Marketing

<https://seranking.com/blog/html-tags-in-seo/>

How to Write a User Story for an API Product

<https://sameera17w.medium.com/how-to-write-a-user-story-for-an-api-product-7af6abd4ad2e>

Firebase App Distribution

<https://firebase.google.com/docs/app-distribution>

Using PlantUML for Creating Clear and Concise Diagrams

<https://connect2grp.medium.com/using-plantuml-for-creating-clear-and-concise-diagrams-2fc621529560>

Big Picture EventStorming

<https://github.com/ddd-by-examples/library/blob/master/docs/big-picture.md>

Design Level EventStorming

<https://github.com/ddd-by-examples/library/blob/master/docs/design-level.md>

Anexos

Anexo A. Estructura para la sección Objetivo del Estudiante (Student Outcome)

El curso contribuye al cumplimiento del Student Outcome ABET:

ABET – EAC - Student Outcome 5

Criterio: La capacidad de funcionar efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos.

En el siguiente cuadro se describe las acciones realizadas y enunciados de conclusiones por parte del grupo, que permiten sustentar el haber alcanzado el logro del ABET – EAC - Student Outcome 5.

Criterio específico	Acciones realizadas	Conclusiones
Trabaja en equipo para proporcionar liderazgo en forma conjunta	Jiménez Rosas, Arturo Eduardo TB1 Morbi vel tortor id eros dictum venenatis id ut dui. Mauris quis tellus sed nunc hendrerit vehicula ac id mauris. Pellentesque volutpat tellus non ligula blandit ullamcorper quis sodales erat. TB2 ... Rodríguez Peña, Jorge Andrés TB1	Fusce cursus dolor et nulla suscipit, sit amet ullamcorper nibh vestibulum. Nam ornare massa eu lobortis porttitor. Nam ut erat feugiat libero pretium semper at ac metus. Sed at eros dapibus, fermentum quam ut, bibendum lacus. Curabitur eget orci eget urna varius commodo. ...
Crea un entorno colaborativo e inclusivo, establece metas, planifica tareas y cumple objetivos.	Jiménez Rosas, Arturo Eduardo TB1 Cras sed diam suscipit, malesuada ex rutrum, fringilla orci. Vestibulum in nunc quis elit suscipit sollicitudin. TB2 ... Rodríguez Peña, Jorge Andrés TB1	Fusce mattis augue a nisl bibendum, quis fringilla neque scelerisque. Vivamus commodo libero eget venenatis imperdiet. Etiam imperdiet quam condimentum velit tempor porttitor. Suspendisse blandit nisl quis mauris vehicula faucibus. ...

Anexo B. Estructura para el Informe de participación

El Final Project Participant Performance Report es un documento en word donde el Team Leader resume la participación de cada integrante y la asigna a cada uno, una calificación entre 0 y 20 (según las alternativas especificadas: 20, 16, 13, 07 o 00).

Estructura del nombre de archivo: *upc-pre-202520- 1asi0572-<NRC>-<startup>-performance-<tbn/tp1/tf1> (.docx y .pdf)*

Adjuntar el archivo en todas las entregas programadas junto al final Project.

Participant Performance Report							
Nombre de Startup		Solvers Squad	Nombre de Producto		Health Advisor		
Entrega		TP1	Team Leader		Jiménez Rosas, Arturo Eduardo		
Ítem	Estudiante	Responsabilidades	Cumplió a tiempo	cumplió a destiempo	cumplió parcialmente	no cumplió (Cero)	Calificación asignada (20 / 16 / 13 / 07 / 0)
1	Jiménez Rosas, Arturo Eduardo	Vivamus commodo libero eget venenatis imperdiet.	X				13
		Etiam imperdiet quam condimentum velit tempor porttitor.		X			
					
		Suspendisse blandit nisl quis mauris vehicula faucibus.				X	
2	Rodríguez Peña, Jorge Andrés	Duis lacinia purus eu urna euismod, at auctor felis pellentesque.	X				20
		Duis porta lectus sit amet tortor aliquam, in dictum magna ullamcorper.	X				
					
		Praesent mattis arcu ut nunc tempus facilisis.	X				
...							
n	Barrera Robles, Luis Miguel	No participó				X	0

Anexo C. Indicaciones para secciones que incluyen Videos

Sección	Características del video	Sobre el contenido	Integración y entrega
Needfinding Interviews	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-<NRC>-<startup>-needfinding-sprint-<n> Formato: .mp4 Duración: En función a cantidad de entrevistas (considerar edición de 3 a 5 minutos por entrevista).	Consolida todas las entrevistas realizadas, incluyendo en cada entrevista títulos con información del entrevistado, el segmento objetivo y la fecha de la entrevista.	Subir el video en Microsoft Stream/Clipchamp en el canal indicado por el docente. Incluir en el informe screenshot del video con enlace al mismo. Incluir redacción de introducción a la sección y registro de cada entrevista, así como el análisis general donde se identifican las variables y los valores representativos a nivel objetivo y subjetivo que servirán de base para la definición de los User Persona. Seguir especificaciones del enunciado.
Prototypes Navigation / Product Navigation	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-<NRC>-<startup>-<prototype/product>navigation-sprint-<n> Formato: .mp4 Duración: En función a cantidad de user flows de aplicaciones (considerar edición de 3 a 5 minutos por aplicación).	Consolida demostración del flujo de navegación del Landing Page y las aplicaciones, priorizando los user flows relacionados con el core business.	Subir el video en Microsoft Stream/Clipchamp en el canal indicado por el docente. Incluir en el informe screenshot del video con enlace al mismo. Incluir redacción de introducción a la sección, resumiendo los flujos de navegación que se incluyen en el video.
Validation Interviews	Cantidad de Videos: 1 Nomenclatura: upc-pre-202520-1asi0572-<NRC>-<startup>-validation-sprint-<n> Formato: .mp4 Duración: En función a cantidad de entrevistas (considerar edición de 3 a 5 minutos por entrevista).	Consolida sesiones y entrevistas de validación en las que usuarios de los segmentos objetivo interactúen con el landing page y con los prototipos de experiencias web y mobile, manifestando sus observaciones. Para cada entrevista se debe incluir títulos con información del entrevistado, el segmento objetivo y la fecha de la entrevista	Subir el video en Microsoft Stream/Clipchamp en el canal indicado por el docente. Incluir en el informe screenshot del video con enlace al mismo. Incluir redacción de introducción a la sección y redacción de registro de cada entrevista, junto con la evaluación de heurísticas de usabilidad, arquitectura de información y diseño inclusivo para la sesión de evaluación. Seguir especificaciones del enunciado y formatos indicados.
About the Product	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-<NRC>-<startup>-about-the-product-sprint-<n> Formato: .mp4 Duración: De 1 a 3 minutos.	Orientación promocional, resumiendo el modelo de negocio, las características y beneficios del producto, incluyendo algunas escenas de interacción con el producto y al menos una opinión por cada segmento objetivo.	Subir el video en Microsoft Stream/Clipchamp y en YouTube. Incluir en el informe screenshot del video con enlace al mismo. Incluir redacción de introducción a la sección. Adicionalmente, incrustar el video en una sección adecuada del Landing Page. Seguir especificaciones del enunciado.
About the Team	Cantidad de videos: 1 Nomenclatura: upc-pre-202520-1asi0572-<NRC>-<startup>-about-the-team-sprint-<n> Formato: .mp4 Duración: En función al contenido (considerar 5 minutos para la sección de retrospectiva del grupo y 1 minuto por cada testimonio de miembro del equipo).	Video que resume el proceso de trabajo realizado, incluyendo escenas de sesiones de trabajo real del equipo, complementando con narración (voz en off) del proceso. Incluye además el testimonio ante cámara de cada participante describiendo actividades realizadas, logro de outcomes y desarrollo de competencias alcanzados.	Subir el video en Microsoft Stream/Clipchamp y en YouTube. Incluir redacción de introducción a la sección, resumiendo el proceso de trabajo y los logros alcanzados por los miembros del equipo. Adicionalmente, incrustar el video en una sección adecuada del Landing Page. Seguir especificaciones del enunciado.

Anexo D. Formato para Evaluación de User Experience según Heurísticas

UX Heuristics & Principles Evaluation Usability – Inclusive Design – Information Architecture

CARRERA	: Ingeniería de Software
CURSO	: Desarrollo de Soluciones IoT
SECCIÓN	: Código de la sección
PROFESORES	: Todos
AUDITOR	: Nombre del Grupo que ejecuta la Sesión de evaluación
CLIENTE(S)	: Nombre de las personas que participan en la sesión

NOTA: Los contenidos de este formato (en color plomo) son referidos a un sitio web de ejemplo. Use este formato como referencia de la estructura que tiene que ser entregada y elimine el contenido en plomo. Coloque su contenido con color azul.

SITE o APP A EVALUAR:

Nombre de App

TAREAS A EVALUAR:

El alcance de esta evaluación incluye la revisión de la usabilidad de las siguientes tareas:

- 1. Registro de un usuario nuevo*
- 2. Publicación de un post*
- 3. Búsqueda de un paquete turístico*
- 4. Reserva de un viaje*
- 5. Cancelación de una reserva*
- 6. Agregar ítems a un pedido*
- 7. Pago de un paquete turístico*
- 8. Etc.*

No están incluidas en esta versión de la evaluación las siguientes tareas:

- 1. Compartir reservas entre usuarios*
- 2. Presentar ofertas de último minuto*
- 3. Administrar puntos del programa de fidelización de viajes*
- 4. Guardar “alerta de paquetes”*
- 5. Etc.*

ESCALA DE SEVERIDAD:

Los errores serán puntuados tomando en cuenta la siguiente escala de severidad

Nivel	Descripción
1	Problema superficial: puede ser fácilmente superador por el usuario ó ocurre con muy poca frecuencia. No necesita ser arreglado a no ser que exista disponibilidad de tiempo.
2	Problema menor: puede ocurrir un poco más frecuentemente o es un poco más difícil de superar para el usuario. Se le debería asignar una prioridad baja resolverlo de cara al siguiente reléase
3	Problema mayor: ocurre frecuentemente o los usuarios no son capaces de resolverlos. Es importante que sean corregidos y se les debe asignar una prioridad alta.
4	Problema muy grave: un error de gran impacto que impide al usuario continuar con el uso de la herramienta. Es imperativo que sea corregido antes del lanzamiento.

TABLA RESUMEN:

#	Problema	Escala de severidad	Heurística/Principio violada(o)
1	No hay un control que permita regresar a la tienda durante el trámite de compra	3	Usability: Libertad y control del usuario
2	Se repiten constantemente algunas opciones	1	Usability: Consistencia y estándares
3	Imágenes sin atributo "alt"	3	Inclusive Design: Proporciona experiencias comparables
4	Incluye un botón "Ver más" pero no existe contenido al que dirigirse	3	Information Architecture: Is it usable?
5	No incluye información de los planes de servicio	2	Information Architecture: Is it findable?
...	...		

Etc.

DESCRIPCIÓN DE PROBLEMAS:

PROBLEMA #1: No hay un control que permita regresar a la tienda durante el trámite de compra

Severidad: 3

Heurística violada: Usabilidad - Libertad y control del usuario

Problema:

Al momento de ingresar nuestros datos, no podemos regresar a la tienda en caso así lo. Una vez el cliente pase al trámite de compra, en caso de que este quiera regresar a la tienda, no hay un botón que lo envíe al inicio de la web, lo cual nos obliga a efectuar el trámite y al momento de elegir más productos, realizar otro, incrementándose así la cantidad de esfuerzo del usuario.

(Incluir además una captura de pantalla ilustrando el problema).

Recomendación:

La más práctica es que al momento en que queramos realizar dicho trámite, el navegador lo abra en una ventana aparte para no perder los cambios realizados en nuestro carrito de compras.

Anexo E. Errores típicos en la traducción y uso de términos para Ingeniería de Software

La competencia de comunicación efectiva juega un rol fundamental en los procesos de Ingeniería de Software. El ingeniero de software debe comunicarse empleando códigos apropiados de acuerdo con la audiencia a la que se dirige, sea a través de la preparación de informes técnicos, presentaciones en proyectos colaborativos, demostraciones, entre otras actividades. En todos los casos, debe validar los atributos de calidad antes de la entrega o presentación, así como validar que el mensaje llegue satisfactoriamente.

La tabla a continuación resume los errores más comunes en la traducción de términos en el área de ingeniería de software. En general, los errores de traducción se consideran errores de comunicación y son objeto de penalización.

Término original	Traducción correcta	Traducción incorrecta	Explicación
<i>Library</i> ⁵	Biblioteca	Librería	Error común producto de la semejanza fonética, pero biblioteca (library) y librería (bookstore) tienen significados diferentes.
<i>Requirement</i> ⁶	Requisito	Requerimiento	Error común debido a la semejanza fonética, sin embargo, requisito (requirement) y requerimiento (request) tienen diferente significado.
<i>Request</i> ⁷	Solicitud / Petición / Requerimiento	Requisito	Consecuencia de la traducción incorrecta de requirement.
<i>Application</i> ⁸	Aplicación	Aplicativo	<i>Aplicativo</i> en español no existe como sustantivo. <i>Aplicativo</i> es un sustantivo solo en portugués.
<i>Elicit</i> ⁹	Obtener	Elicitar	Elicitar no existe en español.

⁵ Vea la traducción de *library* en <https://www.collinsdictionary.com/dictionary/english-spanish/library>.

⁶ Vea la traducción de *requirement* en <https://www.collinsdictionary.com/dictionary/english-spanish/requirement>.

⁷ Vea la traducción de *request* en <https://www.collinsdictionary.com/dictionary/english-spanish/request>.

⁸ Vea la traducción de *application* en <https://www.collinsdictionary.com/dictionary/english-spanish/application>.

⁹ Vea la traducción de *elicit* en <https://www.collinsdictionary.com/dictionary/english-spanish/elicit>.

También se considera un error el uso de forma oral o escrita de traducciones incorrectas resultado de la mutación de términos originales en inglés, de uso típico en ingeniería de software. Términos como *deploy* (desplegar), *deployment* (despliegue), *test* (probar/prueba, según el contexto de uso), *testing* (pruebas) por ejemplo, son de uso típico en ingeniería de software.

En la comunicación oral o escrita, en cualquier medio, solo se permite el uso de términos en idioma original o la traducción correcta de los mismos. Al igual que el caso de traducción con el término erróneo, no se admite el uso de mutaciones como por ejemplo *deployar*, *testear*, *comitear/commitear*, entre otras. Dichos errores se consideran errores de comunicación son objeto de penalización.

También se considera un error de comunicación el uso indiscriminado de términos como *apartado*¹⁰ para evitar referirse a elementos identificables con un término más adecuado, tanto en descripción de la interfaz de usuario (en reemplazo de elementos reconocibles como view, section, dialog, toolbar, sidebar entre otros), como componentes de artefactos elaborados (p.ej. en reemplazo de términos como capítulo, anexo, sección, cuadro, tabla).

Igualmente se considera un error de comunicación, el uso del término *metodología*¹¹ para referirse a métodos, marcos de trabajo, técnicas, aproximaciones u otros conceptos similares (p.ej. Scrum no es una *metodología*, es un marco de trabajo ágil. Referirse a Scrum como *metodología* es un error).

¹⁰ Ver definición de *apartado* en <https://dle.rae.es/apartado>.

¹¹ Ver definición de *metodología* en <https://dle.rae.es/metodología>