

CAPSTONE PROJECT

I. INFORMACION GENERAL

CURSO	: Evolución de Software
CÓDIGO	: SI431
CICLO	: VI
SEMESTRE	: 2020-2
SECCIONES	: Todas
CRÉDITOS	: 4
SEMANAS	: 16
HORAS	: 3 horas (teoría) semanales
PROGRAMA	: Ingeniería de Software

II. LOGRO DEL CURSO

Al terminar el curso, el estudiante diseña y construye una versión completamente operativa de una solución de software aplicando las buenas prácticas de la Gestión de Configuración de Software como parte del ciclo de vida de desarrollo de software.

ABET – EAC - Student Outcome 2: La capacidad de aplicar el diseño de ingeniería para producir soluciones que satisfagan necesidades específicas con consideración de salud pública, seguridad y bienestar, así como factores globales, culturales, sociales, ambientales y económicos.

III. OBJETIVO DEL TRABAJO

Evolución de software es un curso teórico-práctico de especialidad de la carrera de Ingeniería de Software. Tiene como objetivo cubrir conceptos relacionados a la Gestión de Configuración de Software. Dado que dicho término ha evolucionado en el tiempo, es necesario adaptarse a las nuevas necesidades de la industria, por ello además de los conceptos clásicos de Gestión de Configuración, el contenido se complementa con tópicos propios del proceso de evolución del productos, tales como integración continua y despliegue automatizado de software.

Este trabajo tiene por objetivo elaborar una propuesta de diseño de software y la implementación de un subconjunto de requisitos para un producto de software que brinde soporte para un modelo de negocio, consistente en una solución de software orientada a satisfacer necesidades identificadas para un público objetivo, considerando el impacto de la misma en el contexto global, económico, ambiental y social en el cual operaría, haciendo uso de las herramientas y tecnología utilizadas en el curso, aplicando conceptos, técnicas, patrones y buenas prácticas revisados en clase ó resultado de la investigación.

IV. COMPONENTES

Startup Profile

Incluye la descripción de startup, perfiles de los miembros del equipo.

Solution Profile

La introducción 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. Esto 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.

La parte I contiene a la solución actual y todos los artefactos de diseño, análisis, construcción, test de software.

La parte II aplica la gestión de la configuración de software, considerando la administración del cambio a una funcionalidad existente o la inclusión de una nueva funcionalidad

Software Configuration & Change Management

Release Management

Objetivo del Estudiante (Student Outcome)

Cada participante del equipo debe sustentar evidencia de las actividades realizadas en el trabajo final han ayudado a desarrollar cómo las dimensiones del student outcome. Por ello en esta sección debe haber una subsección por cada alumno donde éste describa por escrito la relación entre el outcome, sus dimensiones y el trabajo que ha realizado. Esto se complementa con lo reflejado en los testimonios expuestos que forman parte del video *About The Team*.

A manera de referencia se incluye los criterios o aspectos específicos que corresponden al **ABET – EAC - Student Outcome 2**.

Diseña productos o componentes en ingeniería de software que satisfacen los requerimientos.

Diseño optimizado de componentes al nivel detallado o bajo nivel, empleando estándares, principios y buenas prácticas de diseño propios de la profesión que mejoran su rendimiento.

Diseña la arquitectura de un producto escalable de ingeniería satisfaciendo los principales atributos de calidad.

Utiliza metodologías de acuerdo con las características propias de la solución buscando su optimización.

Diseña proyectos que permiten la implementación de soluciones en ingeniería de software considerando el impacto en la organización.

Diseña un proyecto de desarrollo de soluciones en ingeniería de acuerdo con las restricciones establecidas identificando y mitigando riesgos.

Sustenta el diseño de un proyecto de desarrollo de soluciones en ingeniería cumpliendo estándares y buenas prácticas, además de considerar el impacto del proyecto en el alcance de otros proyectos afines en ejecución.

Diseña y ejecuta los procesos relacionados al desarrollo y mantenimiento de la solución de software en ingeniería considerando el impacto en la organización.

Adapta y establece procesos relacionados al desarrollo del producto de software. Asegura que las variaciones necesarias a los procesos y metodología se encuentren completamente reflejados en el Plan del Proyecto o las necesidades de la organización en la que trabaja a largo plazo.

Adopta procesos de acuerdo con los requisitos y características propias del proyecto proyectando posibles necesidades futuras.

V. EQUIPO DE TRABAJO

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 las implementaciones de experiencias web y móviles, así como la implementación del landing page, evidenciando el desarrollo de las competencias objetivo de este curso y adquiridas en cursos previos.

VI. PROCESO DE TRABAJO

El proyecto se elaborará bajo un enfoque ágil, basado en un proceso enmarcado por los enfoques de Design Thinking, Lean y Agile, iterativo e incremental. El grupo llevará una bitácora en video de sus actividades, lo cual junto con entrevistas de retrospectiva servirán para la construcción del video About-The-Team.

VII. TECNOLOGÍAS

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

Para la programación de componentes orientados a web development de la aplicación se utilizará Java ó C#. Puede utilizarse frameworks.

Para la programación de la biblioteca / framework

Para elaborar los User Persona se utilizará UXPressia / Xtensio.

Para los Journey Map e Impact Map se utilizará UXPressia.

Para el desarrollo de la propuesta de UI y la simulación de interacción se utilizará Lucidchart / Adobe XD / JustInMind / UXPin.

Para los diagramas de Ingeniería de Software se utilizará Structurizr, Lucidchart, StarUML.

Para el control de proyectos, se utilizará PivotalTracker / Trello.

Para el almacenamiento y control de versiones de código se utilizará GIT gestionado desde Github.

VIII. Características de Documentos

Informe de Proyecto del Curso

Documento en Microsoft Word según template

Carátula

Universidad, carrera, ciclo

Nombre del curso

Sección

Nombre del profesor

"Informe de Trabajo Final"

Nombre del startup

Nombre del producto

Relación de integrantes

Mes y año

Contenido

Tabla de contenidos

Objetivo del Estudiante (Student Outcome)

Introducción

Startup Profile

Descripción de la Startup

Perfiles de integrantes del equipo

Solution Profile

Nombre del producto

Antecedentes y problemática

Parte I: El producto

Capítulo I: Research

Capítulo II: Requirements

Capítulo III: Architecture & Design

Capítulo IV: Implementation

Capítulo V: Implementation & Validation

Parte II: Software Configuration & Release Management

Capítulo VI: Software Integration

Capítulo VII: Software Delivery

Capítulo VIII: Software Deployment

Capítulo IX: Release Management

Conclusiones

Conclusiones y recomendaciones

Video About-The-Team.

Informe de Participación: Proyecto del Curso

Documento donde el coordinador resume la participación de cada integrante y la asigna a cada uno una calificación entre 0 y 20.

Presentación: Proyecto del Curso

Archivo de PowerPoint

IX. Referencias

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

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

5W+2H - Técnica de análisis de problemas

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

Lean UX – Chapter 3

<http://leanuxbook.com/images/leanux-sampler.pdf>

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

<https://www.mountaingoatsoftware.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 4 easy steps?

<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/>

Adobe XD tutorials

<https://helpx.adobe.com/xd/tutorials.html>

Acceptance Criteria in Scrum: Explanation, Examples, and Template

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

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/>

A step by step guide to scenario mapping

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

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/>

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

El curso contribuye al cumplimiento del Student Outcome ABET:

ABET – EAC - Student Outcome 2

Criterio: *La capacidad de aplicar el diseño de ingeniería para producir soluciones que satisfagan necesidades específicas con consideración de salud pública, seguridad y bienestar, así como factores globales, culturales, sociales, ambientales y económicos.*

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 2.

Criterio específico	Acciones realizadas	Conclusiones
Diseña productos o componentes en ingeniería de software que satisfacen los requerimientos.		
Diseña proyectos que permiten la implementación de soluciones en ingeniería de software considerando el impacto en la organización.		
Diseña y ejecuta los procesos relacionados al desarrollo y mantenimiento de la solución de software en ingeniería considerando el impacto en la organización.		