



UTPL
La Universidad Católica de Loja

Modalidad Abierta y a Distancia

Prácticum 4.2: Examen Complexivo

Guía didáctica



Facultad de Ingenierías y Arquitectura

Departamento de Ciencias de la Computación y Electrónica

Prácticum 4.2: Examen Complexivo

Guía didáctica

Carrera	PAO Nivel
▪ <i>Tecnologías de la Información</i>	IX

Autores:

Guamán Coronel Daniel Alejandro
Segundo Raymundo Benítez Hurtado



Asesoría virtual
www.utpl.edu.ec

Universidad Técnica Particular de Loja

Prácticum 4.2: Examen Complexivo

Guía didáctica

Guamán Coronel Daniel Alejandro
Segundo Raymundo Benítez Hurtado

Diagramación y diseño digital:

Ediloja Cía. Ltda.

Telefax: 593-7-2611418.

San Cayetano Alto s/n.

www.ediloja.com.ec

edilojacialtda@ediloja.com.ec

Loja-Ecuador

ISBN digital - 978-9942-39-502-3



**Reconocimiento-NoComercial-CompartirIgual
4.0 Internacional (CC BY-NC-SA 4.0)**

Usted acepta y acuerda estar obligado por los términos y condiciones de esta Licencia, por lo que, si existe el incumplimiento de algunas de estas condiciones, no se autoriza el uso de ningún contenido.

Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons **Reconocimiento-NoComercial-CompartirIgual 4.0 (CC BY-NC-SA 4.0)**. Usted es libre de **Compartir – copiar y redistribuir el material en cualquier medio o formato**. **Adaptar – remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos:** **Reconocimiento-** debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatante. **No Comercial-no puede hacer uso del material con propósitos comerciales.** **Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.** No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Índice

1. Datos de información.....	6
1.1. Presentación-Orientaciones de la asignatura	6
1.2. Competencias genéricas de la UTPL.....	6
1.3. Competencias específicas de la carrera.....	6
1.4. Problemática que aborda la asignatura	6
2. Metodología de aprendizaje.....	8
3. Orientaciones didácticas por resultados de aprendizaje.....	11
Primer bimestre.....	11
Resultado de aprendizaje 1.....	11
Contenidos, recursos y actividades de aprendizaje.....	11
Semana 1	11
 Unidad 1. Generalidades del examen complexivo	12
1.1. Aspectos generales del Prácticum 4.2	14
1.2. Dominios académicos definidos para el Prácticum 4.2.....	15
1.3. Esquema del Practicum 4.2	15
1.4. Rol tutorial del Prácticum 4.2.....	18
1.5. Evaluación de las actividades académicas del Prácticum 4.2.....	19
Actividad de aprendizaje recomendada	21
Semana 2 a 3.....	23
 Unidad 2. Taller “Arquitectura de Software”	23
2.1. Estructura del taller.....	23
2.2. Desarrollo del taller.....	24
2.3. Evaluación del taller.....	25
2.4. Ejemplo de caso de estudio	25
Actividad de aprendizaje recomendada	25
Autoevaluación 1	27
Semana 4 a 5.....	33
 Unidad 3. Taller “Arquitectura Empresarial”	33
3.1. Estructura del taller.....	33
3.2. Desarrollo del taller.....	34

3.3. Evaluación del taller.....	34
3.4. Ejemplo de caso de estudio	34
Actividad de aprendizaje recomendada	35
Autoevaluación 2	37
Semana 6 a 7	39
Unidad 4. Taller “Desarrollo de aplicaciones empresariales”	39
4.1. Estructura del taller.....	39
4.2. Desarrollo del taller.....	40
4.3. Evaluación del taller.....	40
4.4. Ejemplo de caso práctico	41
Actividad de aprendizaje recomendada	41
Autoevaluación 3	42
Semana 8	48
Actividades finales del bimestre.....	48
Actividad de aprendizaje recomendada	49
Segundo bimestre	50
Contenidos, recursos y actividades de aprendizaje.....	50
Semana 9 a 15.....	50
Unidad 5. Evaluación del Examen Complexivo oral y escrito.....	50
5.1. Examen Complexivo oral y escrito.....	51
5.2. Rúbrica de Examen Complexivo oral	54
5.3. Ejemplo de caso de estudio a ser evaluado	55
Actividad de aprendizaje recomendada	55
Semana 16	55
Actividades finales del bimestre.....	55
4. Solucionario	57
5. Referencias bibliográficas	62
6. Anexos	67



1. Datos de información

1.1. Presentación-Orientaciones de la asignatura



1.2. Competencias genéricas de la UTPL

- Orientación a la investigación e innovación.

1.3. Competencias específicas de la carrera

- Construir modelos específicos de ciencias de la computación, mediante esquemas matemáticos y estadísticos, para propiciar el uso y explotación eficiente de datos e información.

1.4. Problemática que aborda la asignatura

El Examen Complexivo es una opción de titulación que pertenece a la Unidad de Integración Curricular. En la asignatura Prácticum 4.2 el estudiante debe continuar con el proceso de auto-preparación y autoestudio de los fundamentos teóricos, para proponer y exponer ante un

tribunal evaluador conformado por docentes universitarios, una solución fundamentada y documentada del estudio de caso o contexto empresarial que viene trabajando desde el Prácticum 4.1.

La asignatura Prácticum 4.2 se desarrolla a través de talleres prácticos, en los cuales el estudiante debe complementar y reforzar su propuesta de solución, acorde a los dominios académicos tales como: arquitectura empresarial, arquitectura de software y desarrollo de aplicaciones empresariales. El objetivo de los talleres es preparar al estudiante para que refuerce los fundamentos teóricos y las habilidades prácticas necesarias para rendir el Examen Complexivo.

El Prácticum 4.2 es una asignatura compuesta de dos partes, (i) actividad práctico - experimental y (ii) Examen Complexivo. En la primera parte, el estudiante debe utilizar técnicas de autoaprendizaje adquirido durante la formación académica, para aplicar los fundamentos teóricos en cada uno de los talleres prácticos. En la segunda parte, el estudiante, luego de cursar y aprobar todos los talleres, está habilitado para presentarse a rendir el Examen Complexivo.

El Examen Complexivo está compuesto de una parte oral y otra escrita. Este examen no se convierte en una medición del nivel de memorización de conceptos por parte del estudiante, sino que se enfoca en evaluar la capacidad que cada uno tiene para planificar, analizar, diseñar e implementar diversas soluciones acordes a los dominios académicos.

Recuerde que, para alcanzar el éxito en el desarrollo de la asignatura y en Examen Complexivo, es necesario complementar los fundamentos teóricos estudiados durante su formación académica, solventar sus dudas en los talleres, utilizar estrategias de estudio y revisar bibliografía complementaria disponible de forma digital en la Web. Mucha suerte y el mejor de los éxitos.



2. Metodología de aprendizaje

Con el objetivo de aportar al logro de los resultados de aprendizaje definidos para la asignatura, se utiliza el proceso metodológico por indagación y el proceso metodológico basado en casos.

Por un lado, la indagación permite que el estudiante, a través de su participación activa en cada uno de los talleres, investigue, analice y utilice el pensamiento crítico, para proponer y exponer soluciones ante un problema, usando para ello las bases conceptuales acordes a los dominios académicos planificados en la asignatura.

Por otro lado, el proceso metodológico, basado en casos, tiene como objetivo que los estudiantes asuman la responsabilidad de su aprendizaje mediante la participación activa y constante en el análisis, diseño y solución de problemas. Esta metodología está centrada en el estudiante, por lo que, el docente cumple el rol de facilitador del aprendizaje.

La combinación del proceso metodológico por indagación y el proceso metodológico basados en casos, permitirán que los estudiantes logren los resultados de aprendizaje mediante la investigación reflexiva y, sobre todo, analicen y propongan soluciones a problemáticas enmarcadas en situaciones reales. Por lo tanto, la experiencia y sus conocimientos dentro de TI son primordiales en esta asignatura.

Con el fin de aportar al logro de los resultados de aprendizaje, se sugiere utilizar las siguientes estrategias de estudio:

- Leer la presente guía didáctica, la cual es solamente una ruta en términos de dominios académicos y material bibliográfico. Por ende, le puede conducir a revisar material adicional de forma impresa, recursos web, recursos educativos abiertos o trabajos de investigación escritos en español o inglés.
- Ser cuidadoso con la organización de su tiempo. Usted necesita crear un plan de estudio y un horario como referencia; recuerde no hay tiempo que perder.

- Prestar atención a temas nuevos y recordar temas ya estudiados, tome sus libros, guías, evaluaciones y autoevaluaciones de ciclos anteriores y reviselos nuevamente, esto le ayudará mucho para prepararse con éxito para el Examen Complexivo.
- Las técnicas de lectura y escritura forman parte del proceso enseñanza – aprendizaje. Tómese el tiempo adecuado para leer y posterior a ello resolver los ejercicios teórico-prácticos que se proponen en los talleres; para ello realice resúmenes, nuevas preguntas, diagramas de alto nivel, utilice técnicas y herramientas, con la finalidad de buscar información detallada que apoyen en su resolución.
- CANVAS es el Entorno Virtual de Aprendizaje (EVA) que se utiliza para el desarrollo del curso de la asignatura, por lo tanto, su revisión y participación semanal en el mismo le permitirá apoyarse en documentación, ejemplos y guías que solventarán sus inquietudes.
- Cuando tenga dudas, comentarios o preguntas, utilice los diferentes medios como: sesiones de zoom, correo electrónico, EVA, u otros para ponerse en contacto con su profesor tutor y los tutores de los talleres.
- En el desarrollo de esta asignatura, actividades como video colaboración, foros y actividades dentro de cada uno de los talleres, se programarán durante el periodo académico en fechas que el docente tutor las comunicará para su participación activa.
- Desarrolle las actividades que se proponen en esta guía didáctica y en cada uno de los talleres, ello permitirá prepararse para el Examen Complexivo.
- Es necesario que, para cumplir con las actividades de aprendizaje autónomo, práctico experimentales y entregables que exige Prácticum 4.2, revise la planificación para el trabajo del alumno (primero y segundo bimestre).
- En el contexto del Examen Complexivo (oral y escrito) y previo a su aprobación, la resolución de los aspectos prácticos y las respuestas a preguntas de conocimiento sobre los dominios académicos de la carrera, serán presentados ante un tribunal conformado por un grupo de docentes, el cual calificará el desempeño del estudiante con el rigor académico correspondiente.

- Finalmente, no dude en contactarse con los profesores tutores encargados del desarrollo de la asignatura y los talleres dentro de Prácticum 4.2.

Esperamos que todas las sugerencias expuestas les permitan completar con éxito su plan de formación en Tecnologías de la Información.



3. Orientaciones didácticas por resultados de aprendizaje



Primer bimestre

Resultado de aprendizaje 1

- Demuestra el manejo integral de las competencias adquiridas a lo largo de su formación profesional asumiendo el compromiso ético.

El resultado de aprendizaje de la asignatura lo alcanzará durante el desarrollo de los talleres y actividades planificadas, que serán la base para finalizar con éxito su preparación y se encuentre apto para rendir el Examen Complexivo oral y el Examen Complexivo escrito.

Contenidos, recursos y actividades de aprendizaje



Semana 1

La asignatura Prácticum 4.2 está planificada para desarrollarse en dos partes. Le invitamos a conocer la estructura y las estrategias de estudio.

En la primera parte el estudiante debe complementar y reforzar su propuesta de solución, en base al estudio de caso o contexto trabajado en Prácticum 4.1, a través de los talleres y el desarrollo de las actividades práctico – experimentales, acorde a los dominios académicos de Tecnologías de la Información que un profesional en formación debería conocer y aplicar.

En la segunda parte, el estudiante posterior a cursar y aprobar los talleres y actividades planificadas en el Prácticum 4.2, estará en capacidad de presentarse a rendir el Examen Complexivo.

Empezaremos reforzando los fundamentos teóricos referentes a los dominios académicos de Ingeniería de Software, Arquitectura de Software, Gestión de TI, Arquitectura Empresarial y desarrollo e implementación de aplicaciones empresariales de software.

En esta primera semana ponemos a su consideración las generalidades del Examen Complexivo como opción de titulación, la cual forma parte de la Unidad de Integración Curricular, para luego continuar con la explicación de los dominios académicos que reforzaremos en cada taller.

Unidad 1. Generalidades del examen complexivo

Prácticum 4.2 es una asignatura que tiene como objetivo la evaluación de las competencias del perfil profesional del Ingeniero en Tecnologías de la Información, definidos en el perfil de egreso de la carrera, los mismos que se encuentran organizados por áreas de formación o dominios académicos.

Entre las competencias del perfil profesional del Ingeniero en Tecnologías de la Información constan las siguientes:

- Administrar los servicios de tecnologías de información de la organización utilizando buenas prácticas de la industria, asegurando la continuidad operacional del negocio
- Construir modelos específicos de ciencias de la computación mediante esquemas matemáticos y estadísticos, para propiciar el uso y explotación eficiente de datos e información.
- Implementar aplicaciones de baja, mediana y alta complejidad, integrando diferentes herramientas y plataformas para dar solución a requerimientos de la organización.
- Diseñar aplicaciones de software que permitan, mediante técnicas avanzadas de modelado, dar solución a los requerimientos del cliente utilizando estándares de la industria.
- Modelar procesos de negocio utilizando técnicas y marcos de referencia para identificar problemas, oportunidades de mejora y proponer alternativas que permitan dar soporte a la estrategia del negocio.

- Asegurar la calidad tanto de los productos como de los procesos en los proyectos informáticos, utilizando buenas prácticas definidas por la industria para garantizar sistemas eficientes y negocios rentables.
- Gestionar la implementación de soluciones de negocio, mediante la ejecución de proyectos de TI que cumplan adecuadamente los requisitos especificados por la organización.
- Define y gestiona políticas, normas y procedimientos, mediante el uso de estándares y marcos de referencia, para promover el alineamiento estratégico entre objetivos de negocio y objetivos de TI
- Implementar mecanismos de seguridad física y lógica en los sistemas organizacionales, mediante el uso de estándares y marcos de trabajo internacionales que garanticen la correcta operación del negocio.
- Desarrollar métodos alternativos de acceso a las TI, mediante la implementación de técnicas de interacción humano – computador para propiciar el uso inclusivo de la tecnología.
- Analizar la influencia que las nuevas tecnologías han generado en la estructura organizacional, modelos y procesos de negocio y su relación en un entorno globalizado, para conocer el valor económico que las empresas pueden alcanzar a través del uso del Internet.
- Define y gestiona políticas, normas y procedimientos, mediante el uso de estándares y marcos de referencia para promover el alineamiento estratégico entre objetivos de negocio y objetivos de TI
- Implementar mecanismos de seguridad física y lógica en los sistemas organizacionales, mediante el uso de estándares y marcos de trabajo internacionales que garanticen la correcta operación del negocio.
- Desarrollar métodos alternativos de acceso a las TI, mediante la implementación de técnicas de interacción humano – computador para propiciar el uso inclusivo de la tecnología.
- Analizar la influencia que las nuevas tecnologías han generado en la estructura organizacional, modelos y procesos de negocio y su relación en un entorno globalizado, para conocer el valor económico que las empresas pueden alcanzar a través del uso del Internet.

Estas competencias han sido adquiridas por el estudiante durante el curso y aprobación de cada una de las asignaturas declaradas en el Plan Formativo de la Carrera de Tecnologías de la Información. En los talleres dentro de las asignaturas Prácticum 4.1 y Prácticum 4.2, acorde a los dominios académicos, se hace énfasis en conceptos, metodologías, estándares y herramientas que le permitan al estudiante aplicarlos en un escenario de estudio o contexto empresarial, con el objetivo de proponer una solución fundamentada y documentada ante un problema o necesidad real.

1.1. Aspectos generales del Prácticum 4.2

Según el Reglamento de Régimen Académico del 2020 (CES, 2020), dentro de la unidad de integración curricular, el estudiante debe validar las competencias profesionales para abordar situaciones, necesidades, problemas, dilemas o desafíos de la profesión dentro de un contexto; esto mediante la aprobación de una de las siguientes opciones:

- Opción 1: Desarrollo de un trabajo de integración curricular, o
- Opción 2: Aprobación de un examen de carácter complexivo.

El estudiante que seleccione la “Opción 2: Aprobación de un examen de carácter complexivo” debe prepararse en dos períodos académicos, cursando y aprobando las asignaturas Prácticum 4.1 y Prácticum 4.2, de manera que demuestre la adquisición de los conocimientos y su aplicación práctica a lo largo de su formación académica y con ello alcanzar su titulación.

El Prácticum 4.2 está dividido en dos bimestres. En el primer bimestre, en cada taller planificado se desarrolla casos de estudios o contextos empresariales asociados a los dominios académicos, en el segundo bimestre se desarrolla el Examen Complexivo oral y el Examen Complexivo escrito, todo esto con rigor académico, de manera que se evidencie por parte del estudiante el logro de las competencias específicas declaradas en el perfil de egreso de la carrera.

1.2. Dominios académicos definidos para el Prácticum 4.2

En la asignatura de Prácticum 4.2 el estudiante se prepara para presentarse a rendir el Examen Complexivo oral y el Examen Complexivo escrito como una de las actividades finales de la asignatura. La evaluación de la asignatura está elaborada acorde los dominios de académicos definidos para esta opción de titulación, entre los que constan:

Dominios académicos definidos para el Prácticum 4.2

1.3. Esquema del Practicum 4.2

Antes de iniciar con el estudio de la asignatura Prácticum 4.2, es necesario que tenga en cuenta la siguiente información:

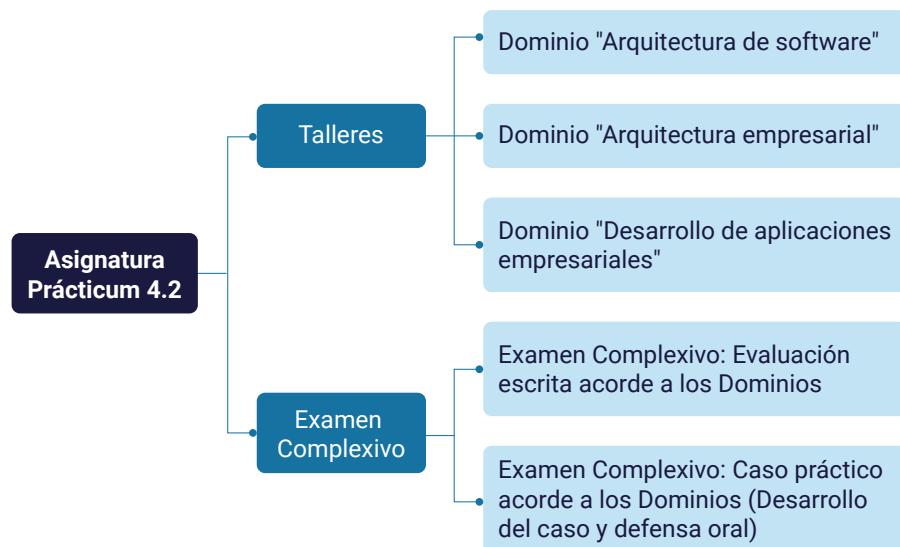
- El estudiante debe tener aprobada la asignatura Prácticum 4.1, además de asignaturas tales como arquitectura de software, metodologías de desarrollo, modelado de sistemas, arquitectura empresarial, fundamentos de ingeniería de software y, especialmente, conocer el proceso de desarrollo de software para diseñar, construir e implementar aplicaciones de software web y/o móviles.
- En la presente asignatura es necesario que desarrolle las actividades planificadas en cada taller, con el objetivo de habilitarlo para rendir el Examen Complexivo oral y el Examen Complexivo escrito de forma presencial (o virtual), donde el estudiante necesita conocimientos teóricos y habilidades prácticas, mismas que serán reforzadas en los cursos Prácticum 4.1 y Prácticum 4.2 y evaluadas dentro del área de Tecnologías de la Información.
- Prácticum 4.2 se desarrolla a través de talleres, donde se pone a prueba sus conocimientos a través de una variedad de preguntas teóricas que son implementadas para dar solución al estudio de caso o escenarios prácticos.
- Las temáticas definidas para los dominios académicos que se evaluarán en el Examen Complexivo oral y el Examen Complexivo escrito (al final del segundo bimestre), se expondrán en los talleres distribuidos en el primer y segundo bimestre.

- La presente guía incluye 5 unidades didácticas, tres en el primer bimestre y dos en el segundo bimestre. Por favor, no se sienta presionado ni desmotivado por la cantidad de temáticas a revisar y no se olvide de buscar información adicional en el caso que lo requiera.

El estudiante que cursa la asignatura de Prácticum 4.2 debe cumplir con la ruta de aprendizaje (Ver Figura 1) que incluye la asistencia, participación y cumplimiento de las actividades en cada uno de los talleres. En los talleres debe poner en práctica su capacidad de análisis sobre el estudio de caso con el objetivo de diseñar, construir y proponer una solución que apoyada en el fundamento teórico y práctico le permitan realizar el Examen Complexivo oral y Examen Complexivo escrito.

Figura 1.

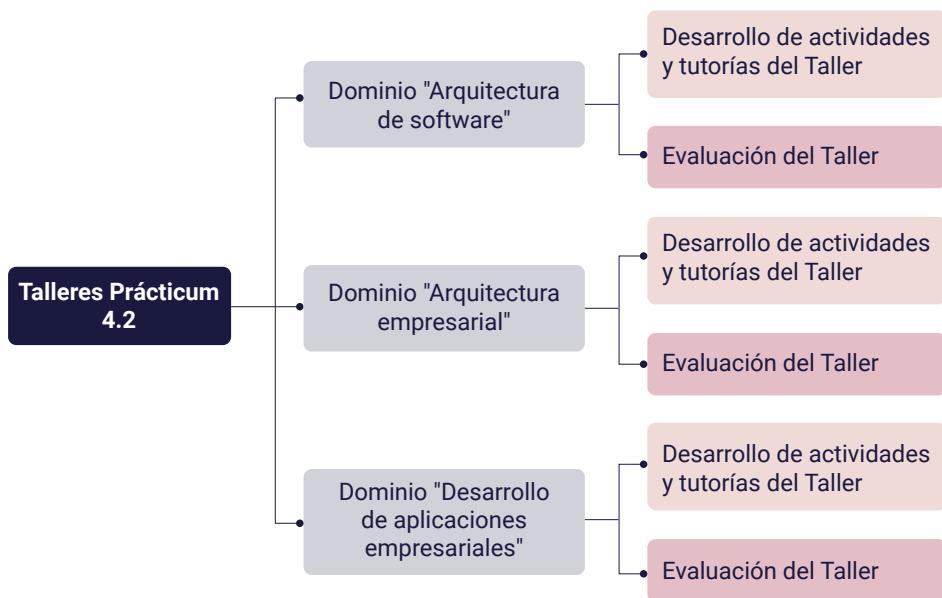
Ruta de aprendizaje de asignatura Prácticum 4.2 - Examen Complexivo



En la figura 2, a continuación, se muestra el esquema para cada uno de los Talleres planificados en la asignatura. Recordarle que por el momento se han definido 3 dominios académicos: Arquitectura Empresarial, Arquitectura de Software y Desarrollo de aplicaciones empresariales.

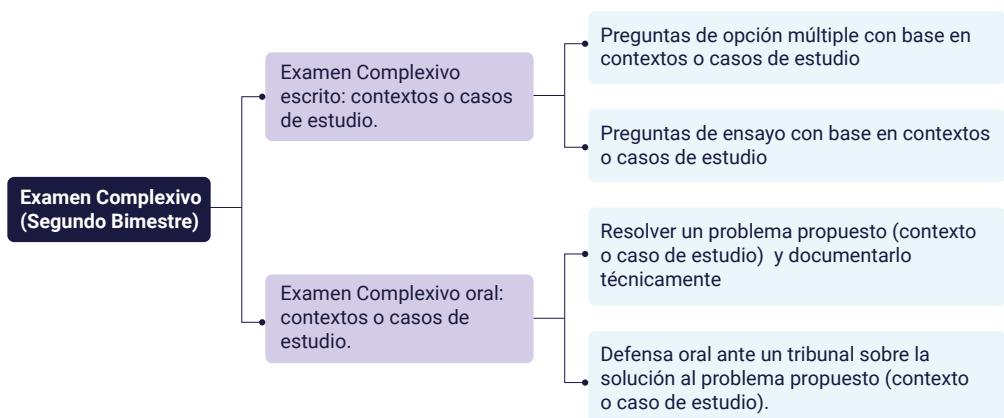
Figura 2.

Talleres Prácticum 4.2 (Dominios y actividades)



En la Figura 3 se muestra el esquema que tendrá el Examen Complexivo (tomado al final del segundo bimestre), mismo que está compuesto por el Examen Complexivo oral y el Examen Complexivo escrito. En la parte escrita, el estudiante debe presentarse a rendir una evaluación que está conformada por preguntas y respuestas de opción múltiple, donde, el tipo de preguntas no se basa solamente en conceptos, sino que le presenta un contexto de un escenario problema y sobre el cual luego de analizarlo puede seleccionar la opción u opciones que sean correctas. En la parte oral, el estudiante debe exponer su propuesta de solución ante un problema propuesto (contexto o caso de estudio), el mismo que ha sido documentado técnicamente. Con todos estos insumos, el estudiante puede presentarse la defensa oral ante un tribunal.

Figura 3.
Esquema del Examen Complexivo



Es importante mencionar que, una vez aprobada todas las actividades planificadas (Talleres, Examen Complexivo oral y Examen Complexivo escrito), el estudiante aprueba la asignatura Prácticum 4.2 y con ello aprueba también el Examen Complexivo como opción de titulación elegida de la Unidad de Integración Curricular de la carrera de Tecnologías de la Información.

Con lo expuesto anteriormente, le invitamos a conocer el rol de los docentes, estudiantes y las actividades académicas planificadas para la presente asignatura.

1.4. Rol tutorial del Prácticum 4.2

Los talleres y el Examen Complexivo oral y escrito (planificado para el final del segundo bimestre) acorde a los dominios académicos, son preparados y tutelados por un conjunto de docentes. Para ello existe la figura de docente principal y uno o varios tutores de talleres acordes al dominio académico. Veamos los roles y actividades en el recurso a continuación.

Roles

1.5. Evaluación de las actividades académicas del Prácticum 4.2

En la Tabla 1 se muestran las actividades académicas a cumplir en la presente asignatura.

Tabla 1.

Actividades académicas

Actividades académicas	Recursos de aprendizaje
PRIMER BIMESTRE	
Aprendizaje en contacto con el docente	Actividades asíncronas en línea (recurso: foro, anuncios académicos).
	Actividades síncronas en línea (recurso: tutorías, chat, video colaboración).
Aprendizaje práctico experimental	Entregables por cada taller (Ensayos, análisis de casos, actividades en laboratorios virtuales y/o presenciales, uso de herramientas).
Aprendizaje autónomo	Autoevaluaciones, evaluaciones parciales y presenciales, lectura de guía didáctica y otros documentos de los talleres (Revisión de videos, audios, presentaciones y Recursos Educativos Abiertos).
SEGUNDO BIMESTRE	
Aprendizaje práctico experimental y Aprendizaje autónomo	Examen Complexivo oral (Resolver un problema propuesto, contexto o caso de estudio, documentar técnicamente y sustentar oralmente ante un tribunal dicha solución).
	Examen Complexivo escrito (Preguntas de opción múltiple o de ensayo con base en contextos o casos de estudio).

En el **primer bimestre**, como se muestra en la Tabla 1, cada taller que se desarrolla en la asignatura de Prácticum 4.2 contempla como mínimo las siguientes actividades:

- Aprendizaje en contacto con el docente:
 - Una video colaboración del taller
 - Tutorías semanales del taller (se publicarán los detalles en el EVA)
- Aprendizaje práctico experimental:
 - Tareas teórico – práctico planificadas para el taller

- Aprendizaje autónomo:

- Evaluación del taller teórico – práctico utilizando los estudios de caso

Las actividades descritas anteriormente y que corresponden al primer bimestre, se deben cumplir por cada uno de los talleres en el tiempo de duración planificados para los mismos.

En el **segundo bimestre**, como se muestra en la Tabla 1, el estudiante debe rendir el Examen Complexivo, según el siguiente detalle:

- Aprendizaje práctico experimental y Aprendizaje autónomo:

- Examen Complexivo oral: Desarrollo de un caso de estudio y defensa ante un tribunal.
 - Examen Complexivo escrito: Evaluación escrita con preguntas de opción múltiple o de ensayo.

En este contexto, para aprobar el Examen Complexivo como opción de titulación, se requiere que los estudiantes obtengan como mínimo el 70% de la sumatoria de la parte escrita y oral. En caso de no alcanzar la nota mínima de 7/10 entre la suma de la nota lograda en el Examen Complexivo oral y la del Examen Complexivo escrito, tiene la oportunidad de presentarse a la evaluación de recuperación del Examen Complexivo oral y del Examen Complexivo escrito, hasta una semana después de las fechas determinadas en el calendario académico para la aplicación de la evaluación de recuperación.

Para aprobar la asignatura Prácticum 4.2 la nota mínima, según lo establecido en el Reglamento de Régimen Académico Interno, es el 70% del total de la nota (7/10 puntos), que se obtiene de la sumatoria de las calificaciones ponderadas de cada una de las actividades establecidas en el plan docente: actividades de los talleres en el primer bimestre y Examen Complexivo oral y Examen Complexivo escrito en el segundo bimestre.

Es importante mencionar que, una vez que el estudiante cumple con todas las actividades académicas planificadas en los talleres y aprueba el Examen Complexivo oral y el Examen Complexivo escrito (tomado el final del segundo bimestre en la presente asignatura), como consecuencia, aprueba la asignatura del Prácticum 4.2 y además aprueba el Examen

Complexivo como opción de titulación de la carrera de Tecnologías de la Información.



Actividad de aprendizaje recomendada

Complete y organice en su drive personal, los entregables trabajados en Prácticum 4.1 acorde a los dominios académicos de Arquitectura de Software, Arquitectura Empresarial y Desarrollo de aplicaciones. Tenga en cuenta las [temáticas generales, las herramientas y los estándares](#) sugeridos para completar la actividad previo al inicio de los talleres.

Dominio	Insumos	Entregables Comunes (Talleres)	Temáticas Generales	Herramientas / Estándares
Arquitectura	Caso General Listado de requerimientos funcionales / no funcionales	E01: Diagrama de Contexto E02: Lista de procesos E03: Lista de necesidades generales E04: Documento de visión E05: Registro de interesados E06: Requisitos de usuario, requisitos funcionales, requisitos no funcionales E07: Acta de constitución del proyecto E08: Documento de especificación de requisitos Proceso SCRUM: <ul style="list-style-type: none">▪ Backlog de producto (Temas, Eneas, Historias, Tareas)▪ 4 +1 View (Krutch)▪ Escenarios▪ Casos de Uso▪ Clases▪ Componentes▪ Despliegue Arquitectura Lógica (MVC) y Arquitectura Física (Tiers)	Ingeniería de Software (SDLC + SDLCS) Ingeniería de requisitos (Requisitos de Negocio, Usuario, Sistema) Metodología de desarrollo Ágil (Backlog de producto) Arquitectura de Software (Definición y diseño de una arquitectura lógica y física) Arquitecturas de Software Evolución	Laboratorio Virtual: Enterprise Architect – UML

Dominio	Insumos	Entregables Comunes (Talleres)	Temáticas Generales	Herramientas / Estándares
Desarrollo de Aplicaciones	Caso General Arquitectura de solución (MVC) On-premise y Cloud Modelo Lógico de datos Diagrama de componentes Diagrama de UI	Aplicación (Front – end + Back – end) Modelo físico BDD Gestión de código Git Plan de pruebas (Funcionales) SonarQube (Calidad y Seguridad)	Front end Back end GitHub comandos básicos	Front – end (HTML, CSS3) Cualquier tecnología
Arquitectura Empresarial	Caso General Arquitectura de Solución	Mapa de capacidades empresariales Modelo Lógico de datos Diagrama de componentes Diagramas de UI (Figma)	Puntos de Vista Dominios de arquitectura empresarial Arquitectura de negocio Arquitectura de TI Transición	Laboratorio Virtual: Enterprise Architect – UML



Semana 2 a 3

Una vez que conozca las generalidades del Prácticum 4.2, le invitamos a partir de esta semana a iniciar con el primer taller y desarrollar las actividades planificadas.

Unidad 2. Taller “Arquitectura de Software”

Este taller tiene como objetivo que los estudiantes refuerzen conceptos, estándares, paradigmas, modelos, metodologías, marcos de trabajo y herramientas, que son útiles para gestionar un proyecto y un producto software.

2.1. Estructura del taller

Temáticas de asignaturas:

El presente taller se desarrolla con base en un conjunto de asignaturas y temáticas abordadas en el transcurso de la carrera, estas son:

- Fundamentos de ingeniería de software como se detalla en Abad (2019).
- Ingeniería de requisitos como se detalla en Sucunuta (2017).
- Arquitectura de software como se detalla en Guamán (2021).
- Modelado de sistemas como se detalla en Guamán (2018).
- Gestión de proyectos como se detalla en Abad (2021).

Adicional a ello, en el EVA se publicarán otras temáticas complementarias que deberán ser revisadas por los estudiantes para el desarrollo de las actividades planificadas en el taller.

Material Bibliográfico sugerido:

El material bibliográfico y recursos educativos relacionados con el taller como presentaciones, guías y casos de estudio, serán compartidos por el profesor tutor en el EVA.

2.2. Desarrollo del taller

El presente taller requiere que el estudiante refuerce las bases conceptuales necesarias que le permitan gestionar un proyecto y un producto software, adoptando el rol de Analista de sistemas, Ingeniero de Software y Arquitecto de Software.

Las actividades teórico – prácticas enmarcadas en el dominio de Arquitectura de Software y que deben cumplirse en este taller son:

- Revisar la documentación del taller publicado en el EVA.
- Complementar el estudio de los conceptos que le permitan el desarrollo de los estudios de caso, con material disponible en la Web.
- Participar de las video colaboraciones planificadas dentro del taller.
- Desarrollar todas las actividades síncronas y asíncronas planificadas durante el taller.

Luego de cumplir con todas las actividades planificadas del taller, se deben medir los conocimientos adquiridos por cada estudiante a través de una evaluación teórico/práctica, la misma que se explicará en el siguiente apartado.

2.3. Evaluación del taller

La evaluación del taller “Arquitectura de software” consiste en analizar un caso de estudio alineado al dominio académico y proponer una solución de diseño.

La evaluación del taller se aplicará según los lineamientos de la UTPL, en la fecha y horario señalado por el docente tutor, con una duración aproximada de entre 1 y 2 horas.

2.4. Ejemplo de caso de estudio

Como parte del estudio se propone el desarrollo y análisis de estudios de caso prácticos y guiados, los cuales han sido preparados por los profesores tutores de los talleres y están anexos en la guía (Ver [Anexo1](#), [Anexo2](#), [Anexo3](#)). Para la comprensión de los estudios de caso usted necesita revisar conceptos que pertenecen a cada taller cursado en esta asignatura, además de conceptos estudiados en ciclos anteriores.

El estudio de caso detallado en los citados anexos, integra todas las temáticas abordadas en los talleres de la asignatura (“Arquitectura de Software”, “Arquitectura Empresarial” y “Desarrollo de aplicaciones empresariales”).

Esperamos que con el ejemplo de caso de estudio haya reforzado sus conocimientos en torno a las temáticas abordadas en el taller. ¡Buena Suerte!



Actividad de aprendizaje recomendada

Utilizando conceptos de arquitectura de software y herramientas de diseño, represente en alto nivel su propuesta de solución, donde se evidencie que los clientes interactuarán con solución desde un dispositivo web y móvil y que toda su infraestructura donde se encuentra la aplicación, los servicios y la base de datos estarán desplegadas en la nube.

Estimado Estudiante, lo invito a desarrollar la autoevaluación con la que pondrá a prueba los conocimientos utilizados en cada taller, donde se han

incluido conceptos de ingeniería de software, arquitectura de software, metodologías de desarrollo y conceptos usados actualmente por un Ingeniero en Tecnologías de la Información.



Autoevaluación 1

Para dar respuesta a las siguientes preguntas, revise cada ítem y seleccione la respuesta correcta. Recuerde que sus resultados constituyen un reflejo del autoaprendizaje requerido para el examen complejivo.

- 1. ¿Con qué principio de la arquitectura de software se asocia la siguiente definición? Los sistemas deben construirse a partir de componentes (módulos) con alta cohesión y bajo acoplamiento, lo que significa que el sistema debe estar formado o dividirse en diferentes componentes o piezas más simples que están unidas y funcionan juntas de manera eficiente.**
 - a. Abstracción.
 - b. Generalidad.
 - c. Modularidad.

- 2. ¿Con qué principio de la arquitectura de software se asocia la siguiente definición? Es un proceso mediante el cual los diseñadores y arquitectos de software identifican los aspectos relevantes de un problema, ignorando los detalles de implementación de los componentes.**
 - a. Abstracción.
 - b. Generalidad.
 - c. Modularidad.

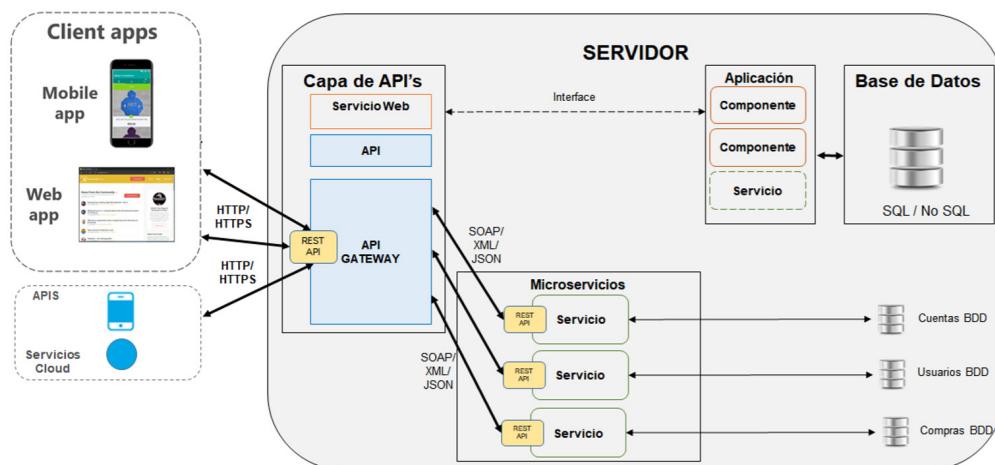
- 3. ¿Cuáles son los modelos de procesos que combinan las metodologías ágiles?**
 - a. Evolutivo – Prototipado.
 - b. Iterativo – Incremental.
 - c. Evolutivo – Iterativo.

- 4. Scrum es una metodología ágil que:**
- a. Utiliza un conjunto de buenas prácticas que permite trabajar colaborativamente en equipo para obtener los mejores resultados posibles a través de la entrega progresiva de valor.
 - b. Utiliza un conjunto de fases y procesos que se deben seguir para obtener los mejores resultados posibles a través de la entrega progresiva de valor.
 - c. Permite construir productos o servicios de *software* de forma ágil y dinámica, que utilizan un conjunto de prácticas para obtener el mayor valor posible.
- 5. Las historias de usuarios son parte de un enfoque ágil que ayuda a cambiar el enfoque de escribir sobre los requisitos al hablar sobre ellos, ¿cuál de las siguientes opciones es correcta?, respecto de las historias de usuario.**
- a. Una descripción en lenguaje natural que representa un requisito de *software*.
 - b. Una descripción en lenguaje técnico que representa un requisito de *software*.
 - c. Una historia narrada por el usuario que describe una característica de *software*.
- 6. DevOps es un conjunto de prácticas, herramientas y filosofía cultural que sirve para automatizar e integrar los procesos que comparten el equipo de:**
- a. Desarrolladores y operaciones.
 - b. Desarrolladores y analistas.
 - c. Desarrolladores y clientes.

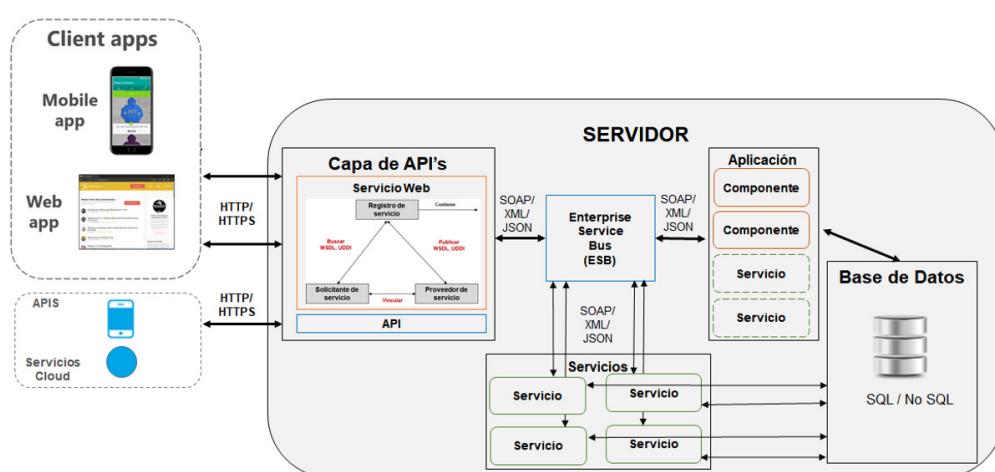
- 7. ¿Cuál de las siguientes opciones es correcta respecto al concepto de integración continua?**
- a. Una práctica de desarrollo de *software*, donde todos los miembros del equipo integran el código con frecuencia y este es verificado de forma automática a través de un servidor de integración.
 - b. Una práctica de desarrollo de *software*, donde todos los miembros del equipo de desarrollo integran las nuevas características de *software* a través de un repositorio de código.
 - c. Un proceso automático que integra el código de los desarrolladores para desplegar las aplicaciones.
- 8. ¿Cuál de las siguientes opciones es correcta?, respecto del concepto de entrega continua.**
- a. Extensión de la integración continua, su objetivo es recuperar el código fuente y evaluarlo para encontrar defectos.
 - b. Se complementa con la integración continua, su objetivo es compilar el código fuente y generar un artefacto binario.
 - c. Su objetivo es recuperar el código fuente, compilarlo y desplegarlo para verificar el funcionamiento de la aplicación.
- 9. La integración continua (CI) y la entrega continua (CD) encarnan una cultura, un conjunto de principios operativos y una colección de prácticas que permiten a los equipos de desarrollo de aplicaciones entregar cambios de código con mayor frecuencia y confiabilidad. La implementación también se conoce como canalización de CI/CD. Por lo tanto, ¿qué garantiza una canalización CI/CD?**
- a. La entrega de *software* rápida y de calidad.
 - b. La entrega de *software* flexible y eficiente.
 - c. La entrega de *software* para implementaciones futuras.

10. La arquitectura de micro-servicios es una aproximación para el desarrollo de software, consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros (normalmente una API de recursos HTTP), ¿cuál de las siguientes figuras representa un escenario en el cual intervienen los micro-servicios?

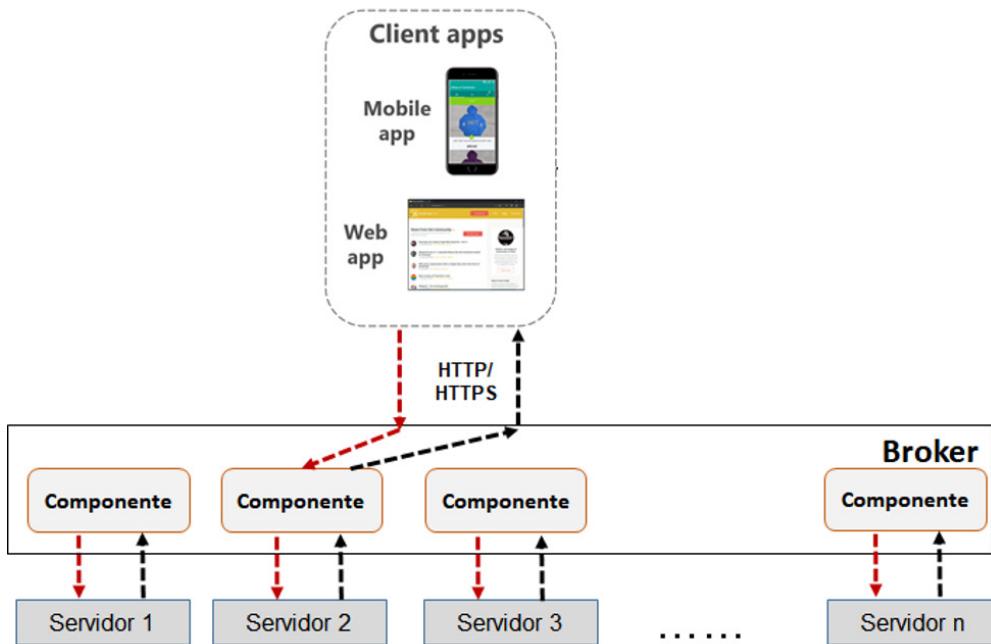
a.



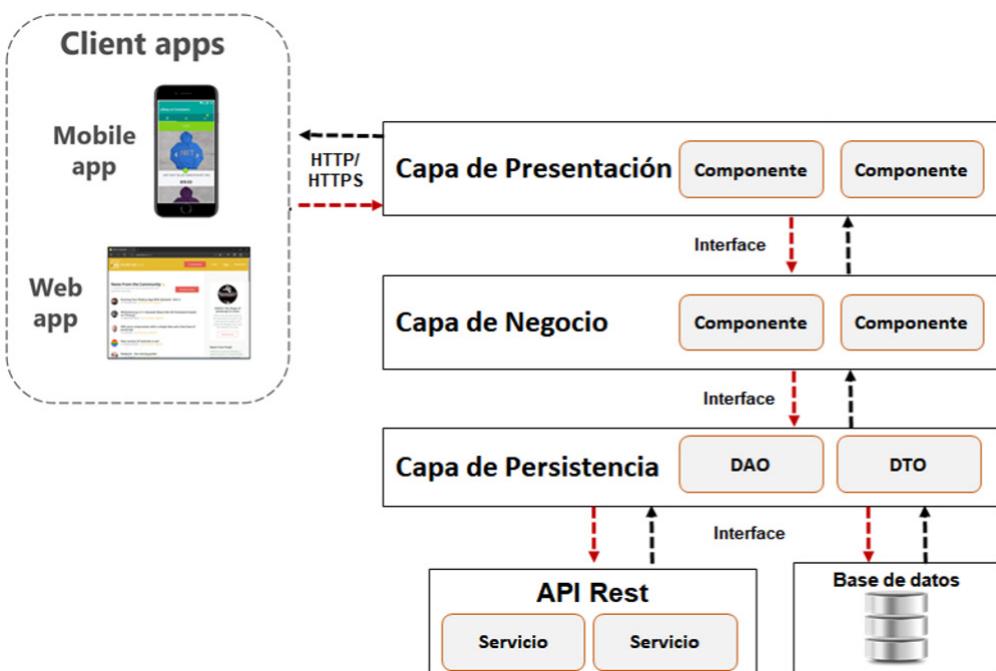
b.



C.



d.



Ir al solucionario

¿Cómo le fue en la autoevaluación?, espero que muy bien. No olvide verificar sus respuestas con las que se encuentran al final de la presente guía didáctica. Si no consiguió un buen resultado, es necesario que revise nuevamente los conceptos que aún no estén claros o persistan dudas. Recuerde interactuar con su tutor y de hacer uso de los medios de comunicación que le brinda la UTPL, esto es muy importante que lo tenga siempre presente.



Semana 4 a 5

¿Qué le pareció el taller 1 donde se reforzaron los conceptos y su aplicación práctica para proponer una solución y documentar una arquitectura de software? Esperamos que conforme avancemos en el desarrollo de la asignatura, afiance los conceptos y sus técnicas de aplicación práctica que todo Ingeniero en Tecnologías de la Información debe conocer. Es momento de continuar con el segundo taller, recuerde que cada taller requiere que se cumplan con las actividades trabajadas previamente.

Unidad 3. Taller “Arquitectura Empresarial”

Este taller tiene como objetivo que los estudiantes refuerzen conceptos y terminología relacionada con la Arquitectura Empresarial, su contexto, frameworks como TOGAF, herramientas y lenguajes de modelado como Archimate, para proponer, diseñar y desarrollar soluciones que integren el negocio y las TIC (Tecnologías de Información y Comunicación).

3.1. Estructura del taller

Temáticas de asignaturas:

El desarrollo del presente taller requiere que revise las temáticas principales de las asignaturas abordadas en el transcurso de la carrera, estas son:

- Planificación estratégica y sistemas de calidad empresarial, como se detalla en Cabrera y Torres (2020).
- Arquitectura Empresarial, como se detalla en Torres y Cabrera (2021).
- Organización y Administración de Infraestructura de TI como se detalla en Cueva y Picoita (2021).
- Gobernanza de Tecnologías de la Información
- Proyectos de Tecnologías de la Información

Adicionalmente a ello, en el EVA se publicarán otros temas complementarios para el desarrollo del taller.

Material Bibliográfico sugerido:

El material bibliográfico y recursos educativos relacionados con el taller como presentaciones, guías y casos de estudio, serán compartidos por el profesor tutor en el EVA.

3.2. Desarrollo del taller

El presente taller tiene como objetivo que el estudiante afiance las bases conceptuales necesarias para el desarrollo de Arquitectura Empresarial en las organizaciones, con base en Frameworks o marcos reconocidos en el medio.

Las actividades teórico – prácticas enmarcadas en el dominio de Arquitectura Empresarial y que deben cumplirse en este taller son:

- Revisar la documentación del taller publicado en el EVA.
- Complementar el estudio de los conceptos que le permitan el desarrollo de los estudios de caso, con material disponible en la Web.
- Participar de las video colaboraciones planificadas dentro del taller.
- Desarrollar todas las actividades síncronas y asíncronas planificadas durante el taller.

Luego de cumplir con todas las actividades planificadas del taller se deben medir los conocimientos adquiridos por cada estudiante mediante una evaluación, la misma que se explicará en el siguiente apartado.

3.3. Evaluación del taller

La evaluación del taller de "Arquitectura Empresarial", consiste en analizar un caso de estudio alineado al mismo y proponer una solución al respecto.

La evaluación del taller se aplicará según los lineamientos de la UTPL, en la fecha y horario señalado por el docente tutor, con una duración aproximada de entre 1 y 2 horas.

3.4. Ejemplo de caso de estudio

Como parte del estudio se propone el desarrollo y análisis de estudios de caso prácticos y guiados, los cuales han sido preparados por los profesores tutores de los talleres y están anexos en la guía ([Ver Anexo1](#), [Anexo2](#),

[Anexo3](#)). Para la comprensión de los estudios de caso usted necesita revisar conceptos que pertenecen a cada taller cursado en esta asignatura, además de conceptos estudiados en ciclos anteriores.

El estudio de caso detallado en los citados anexos integra todas las temáticas abordadas en los talleres de la asignatura ("Arquitectura de Software", "Arquitectura Empresarial" y "Desarrollo de aplicaciones empresariales").

Esperamos que con el ejemplo de caso de estudio haya reforzado sus conocimientos en torno a las temáticas abordadas en el taller.

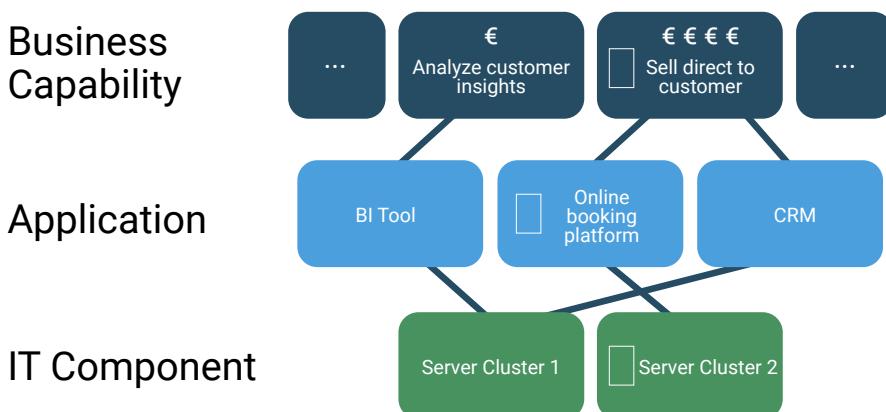
Si aún existen dudas, por favor revise la bibliografía base y complementaria disponible en la web, y/o contacte con su docente/tutor a través de los distintos medios como correo electrónico, plataforma académica virtual (EVA) o línea telefónica.



Actividad de aprendizaje recomendada

Utilizando conceptos de arquitectura empresarial, herramientas y estándares de documentación, represente en alto nivel las capacidades de negocio, aplicación y componentes de TI de su propuesta de solución. Tome como referencia la siguiente figura 4.

Figura 4.
Niveles de Arquitectura Empresarial



Estimado Estudiante, lo invito a desarrollar la autoevaluación con la que pondrá a prueba los conocimientos utilizados en cada taller, donde se han incluido conceptos de arquitectura empresarial que todo Ingeniero en Tecnologías de la Información debe conocer.



Autoevaluación 2

Para dar respuesta a las siguientes preguntas, revise cada ítem y seleccione la respuesta correcta. Recuerde que sus resultados constituyen un reflejo del autoaprendizaje requerido para el examen complejivo.

- 1. Al proceso de concebir, analizar, documentar, comunicar y validar la implementación, la gestión y el mantenimiento de la arquitectura, a través de todo su ciclo de vida, se le llama:**
 - a. Arquitectura.
 - b. Descripción de la arquitectura.
 - c. Punto de vista.
- 2. El punto de vista es un enfoque de visualización del sistema que dicta las convenciones para estructurar, interpretar y analizar:**
 - a. Un modelo.
 - b. Una arquitectura empresarial.
 - c. Una vista.
- 3. Los puntos de vista se clasifican en:**
 - a. Negocio, información, aplicaciones y tecnología.
 - b. Partes interesadas, preocupaciones y modelo.
 - c. Diseño, decisión e información.
- 4. Ayudan a los gerentes en el proceso de toma de decisiones al ofrecer una perspectiva de las relaciones de arquitectura entre dominios.**
 - a. Puntos de vista de decisión.
 - b. Puntos de vista de diseño.
 - c. Puntos de vista de información.
- 5. Las personas son consideradas conductores:**
 - a. Internos.
 - b. Externos.
 - c. Estructura organizativa.

- 6. Cuando la arquitectura empresarial es utilizada para la toma decisiones, la provisión de recursos o las inversiones, el posicionamiento debe centrarse en:**
- a. La planificación estratégica.
 - b. Las TI.
 - c. Arquitectura tecnológica.
- 7. La arquitectura empresarial realiza descripciones de la organización del:**
- a. Estado actual.
 - b. Solo del estado futuro.
 - c. Estado actual y estado futuro.
- 8. Se realiza una búsqueda de oportunidades de mejora en:**
- a. Las arquitecturas as-is.
 - b. Las arquitecturas to-be.
 - c. La transición.
- 9. Los resultados de las arquitecturas contienen descripciones de:**
- a. Arquitecturas to-be.
 - b. Arquitecturas as-is, arquitecturas to-be y planes de transición.
 - c. Arquitecturas to-be y planes de transición.
- 10. ¿Qué elementos no forman parte de la arquitectura de negocio?**
- a. Capacidades empresariales.
 - b. Conductores.
 - c. Sistemas.

[Ir al solucionario](#)



Unidad 4. Taller “Desarrollo de aplicaciones empresariales”

A través del presente taller se espera que los estudiantes, refuerzen los conocimientos que le permitan analizar y utilizar buenas prácticas, para construir e implementar una solución de software empresarial usando tecnologías emergentes (web, móviles, servicios, microservicios). Es importante mencionar, que la solución de software debe ser construida usando los artefactos desarrollados a través de la metodología de desarrollo, en nuestro caso SCRUM, y usando los modelos y diseños de la arquitectura lógica y física que usted propuso y trabajó en los talleres de Arquitectura de Software y Arquitectura Empresarial.

4.1. Estructura del taller

Temáticas de asignaturas:

El presente taller requiere que usted aplique conceptos y buenas prácticas estudiadas en asignaturas y temáticas abordadas en el transcurso de la carrera, estas son:

- Ingeniería de software
- Ingeniería de requisitos
- Metodologías de desarrollo como se detalla en Soto y Correa (2020).
- Desarrollo Web como se detalla en Torres (2019)
- Itinerario (s) Desarrollo, Gestión de la calidad del Software como se detalla en Cueva y Rodríguez (2020).
- Administración de base de datos como se detalla en Morocho et. al. (2020).
- Arquitectura de Software

Adicionalmente a ello, en el EVA se publicarán recursos complementarios para el desarrollo del taller.

Material Bibliográfico sugerido:

El material bibliográfico y recursos educativos relacionados con el taller como presentaciones, guías y casos de estudio, serán compartidos por el profesor tutor en el EVA.

4.2. Desarrollo del taller

El objetivo del taller es afianzar las bases conceptuales necesarias, para el desarrollo e implementación de aplicaciones web que interactúan con bases de datos y que usan estilos, patrones arquitectónicos, estándares y buenas prácticas de desarrollo de software.

Las actividades teórico – prácticas enmarcadas en el desarrollo web y base de datos que deben cumplirse en este taller son:

- Revisar la documentación del taller publicado en el EVA.
- Complementar el estudio de los conceptos que le permitan el desarrollo de los estudios de caso, con material disponible en la Web.
- Participar de las video colaboraciones planificadas dentro del taller.
- Desarrollar todas las actividades síncronas y asíncronas planificadas durante el taller.

Luego de cumplir con todas las actividades planificadas del taller, se deben medir los conocimientos adquiridos por cada estudiante mediante una evaluación, la misma que se explicará en el siguiente apartado.

4.3. Evaluación del taller

La evaluación del taller “Desarrollo de aplicaciones empresariales”, consiste en analizar un caso de estudio alineado al mismo y proponer una solución.

La evaluación del taller se aplicará según los lineamientos de la UTPL, en la fecha y horario señalado por el docente tutor, con una duración aproximada de entre 1 y 2 horas.

4.4. Ejemplo de caso práctico

Como parte del estudio se propone el desarrollo y análisis de estudios de caso prácticos y guiados, los cuales han sido preparados por los profesores tutores de los talleres y están anexos en la guía ([Ver Anexo1](#), [Anexo2](#), [Anexo3](#)). Para la comprensión de los estudios de caso, usted necesita revisar conceptos que pertenecen a cada taller cursado en esta asignatura, además de conceptos estudiados en ciclos anteriores.

El estudio de caso detallado en los citados integra todas las temáticas abordadas en los talleres de la asignatura ("Arquitectura de Software", "Arquitectura Empresarial" y "Desarrollo de aplicaciones empresariales").

Esperamos que, con el ejemplo de caso de estudio, haya reforzado sus conocimientos en torno a las temáticas abordadas en el taller.

¡Muy bien!, hemos completado el estudio de los dominios de la carrera de Tecnología de la Información por medio de los tres talleres, por lo que, hemos culminado con el proceso de preparación para rendir el Examen Complexivo como opción de titulación.

En la siguiente unidad se explicará cómo se desarrollará el Examen Complexivo oral y el Examen Complexivo escrito.



Actividad de aprendizaje recomendada

Elabore un pipeline que permitiría que su propuesta de solución integre conceptos de DevOps, CI/CD para los temas de planificación, construcción, pruebas y despliegue de su solución.

Estimado Estudiante, lo invito a desarrollar la autoevaluación con la que pondrá a prueba los conocimientos utilizados en cada taller, donde se han incluido conceptos de desarrollo de software y bases de datos que todo Ingeniero en Tecnologías de la Información debe conocer.



Autoevaluación 3

Para dar respuesta a las siguientes preguntas revise cada ítem y seleccione la respuesta correcta. Recuerde que sus resultados constituyen un reflejo del autoaprendizaje requerido para el examen complexivo.

- 1. ¿Cuáles de las siguientes operaciones básicas se pueden realizar cuando se escribe el código de una aplicación de software, al momento de trabajar con bases de datos relacionales?**
 - a. Escritura, lectura, actualización, borrado.
 - b. Abrir la base de datos y cerrar la base de datos.
 - c. Conectarse a la base de datos y capturar las excepciones.
 - d. Ninguna de las anteriores.
- 2. La palabra reservada SQL, que permite escribir datos sobre las tablas de una base de datos, es:**
 - a. CREATE.
 - b. UPDATE.
 - c. INSERT.
 - d. DELETE.
- 3. A nivel de programación, dentro del uso de genéricos, ¿cuál de las siguientes declaraciones de listas es correcta?**
 - a. List<Integer> NotasComponentes = new ArrayList<Integer>();
 - b. List<Integer> NotasComponentes = new ArrayList<String>();
 - c. List<String> NotasComponentes = new ArrayList<Integer>();
 - d. List<Integer> int = new ArrayList<String>();

- 4. Una forma básica para el uso de una sentencia de registro INSERT es:**
- a. `INSERT INTO nombreDeTabla (nombreDeColumna1, nombreDeColumna2, ..., nombreDeColumnaN) VALUES (valor1, valor2, ..., valorN);`
 - b. `INSERT INTO nombreDeTabla (nombreDeColumna1, nombreDeColumna2, ..., nombreDeColumnaN);`
 - c. `INSERT INTO (nombreDeColumna1, nombreDeColumna2, ..., nombreDeColumnaN) VALUES (valor1, valor2, ..., valorN);`
 - d. Todas las anteriores.
- 5. ¿Cuál de las siguientes declaraciones de sentencias SQL permiten eliminar los registros de una tabla?**
- a. `DELETE FROM nombre_tabla;`
 - b. `SELECT * FROM nombre_tabla;`
 - c. `UPDATE FROM nombre_tabla;`
 - d. `INSERT FROM nombre_tabla;`
- 6. ¿Cuál es el lenguaje de consulta de bases de datos estándar internacional?**
- a. PHP.
 - b. JAVA.
 - c. API JDBC.
 - d. SQL.

7. Con base en la siguiente sección de código, responda a las inquietudes que se presentan:

```
1. package javaapplication2;  
2. import _____  
3. public class JavaApplication2 {  
4.     public static void main(String[] args) throws Exception {  
5.         Connection conn = getConnection();  
6.         java.sql.Statement sentencia = conn.createStatement();  
7.         java.sql.ResultSet rsResultado = sentencia.executeQuery("select *  
from ESTUDIANTES");  
8.         while (rsResultado.next())  
9. {  
10.     System.out.println(rsResultado.getString("CEDULA"));  
11.     System.out.println(rsResultado.getString("NOMBRES"));  
12.     System.out.println(rsResultado.getString("APELIDOS"));  
13. }  
14.     rsResultado.close();  
15.     sentencia.close();  
16.     conn.close();  
17. }
```

¿Qué paquete debemos importar para el correcto funcionamiento del programa?

- a. java.io.*
- b. java.util.*
- c. java.sql.*;
- d. java.awt.*

8. Con base en la siguiente sección de código responda a las inquietudes que se presentan:

```
1. public static void main(String[] args) throws Exception {  
2.     Connection conn = getConnection();  
3.     java.sql.Statement sentencia = conn.createStatement();  
4.     String sql= "INSERT INTO ESTUDIANTES VALUES  
(7,'1108','BBBBB','XXXXX')";  
5.     int rsResultado = sentencia.executeUpdate(sql);  
6.     sentencia.close();  
7.     sentencia.close();  
8.     conn.close();  
9. }
```

En las columnas de la tabla ESTUDIANTES, ¿el orden en que los valores se almacenan son de tipo de dato?

- a. Integer, String, String, String.
- b. String, String, String, String.
- c. Integer, Integer, String, String.
- d. Todas las anteriores.

9. Con base en la siguiente sección de código responda a las inquietudes que se presentan:

```
1. public static void main(String[] args) throws Exception {  
2.     Connection conn = getConnection();  
3.     java.sql.Statement sentencia = conn.createStatement();  
4.     String sql= "INSERT INTO ESTUDIANTES VALUES  
('7','1108','BBBBB','XXXXX')";  
5.     int rsResultado = sentencia.executeUpdate(sql);  
6.     sentencia.close();  
7.     sentencia.close();  
8.     conn.close();  
9. }
```

En las columnas de la tabla ESTUDIANTES, ¿el orden en que los valores se almacenan son de tipo de dato?

- a. Integer, String, String, String.
- b. String, String, String, String.
- c. Integer, Integer, String, String.
- d. Todas las anteriores.

10. En base a la siguiente sección de código responda a las inquietudes que se presentan:

```
1. public static void main(String[] args) throws Exception {  
2.     Connection conn = getConnection();  
3.     java.sql.Statement sentencia = conn.createStatement();  
4.     String sql= "INSERT INTO ESTUDIANTES VALUES  
(7,'1108','BBBBB','XXXXX')";  
5.     int rsResultado = sentencia.executeUpdate(sql);  
6.     sentencia.close();  
7.     sentencia.close();  
8.     conn.close();  
9. }
```

El resultado que se espera de la instrucción en la línea cinco puede ser:

- a. 100
- b. "100"
- c. "CORRECTO"
- d. true

[Ir al solucionario](#)

Recuerde que, la autoevaluación, más que una medición de conocimientos, es una apreciación de cómo en la práctica aplica los conceptos; no olvide verificar sus respuestas con las que se encuentran al final de la presente guía didáctica. Si no consiguió un buen resultado, es necesario que revise nuevamente los conceptos que aún no estén claros o persistan dudas. Recuerde interactuar con su tutor y de hacer uso de los medios de comunicación que le brinda la UTPL, esto es muy importante que lo tenga siempre presente.



Actividades finales del bimestre

Luego de actualizar y reforzar los conocimientos durante estas 8 semanas de auto-estudio y aprendizaje continuo, seguro ha mejorado su propuesta de solución al estudio de caso. Aspiramos que en cada uno de los talleres se hayan solventado las dudas respecto de la parte conceptual y su aplicación práctica a través de las herramientas donde se planifica, diseña y construye una propuesta de solución. Le recordamos apoyarse en los anexos para que estudie un caso de estudio adicional y que sea la base para proponer más de una solución a un problema.

En los talleres, parte fundamental es que asocie los conceptos teóricos con la práctica en escenarios donde se usan herramientas, buenas prácticas, marcos de trabajo y documentación que involucra las tendencias tecnológicas acordes a los dominios académicos. Recuerde que los conceptos usados en cada uno de los talleres los puede encontrar en cualquier bibliografía de las asignaturas que cursó en ciclos anteriores, pero es responsabilidad de los ingenieros de Tecnologías de la Información llevarlos a la práctica.

Si aún existen dudas, por favor apoyese y consulte bibliografía complementaria y/o contacte con su docente/tutor a través de los distintos medios como correo electrónico, plataforma académica virtual (EVA) o línea telefónica.

Es momento de tomar un descanso y en poco tiempo le animo a que con la misma dedicación y empeño iniciemos el estudio de las unidades que corresponden al Segundo Bimestre, como paso final para el examen complejivo ¡Buena Suerte!



Actividad de aprendizaje recomendada

Revisar conceptos que son usados en los dominios académicos de ingeniería y arquitectura de software, arquitectura empresarial y desarrollo de aplicaciones empresariales.



Segundo bimestre

Al iniciar este segundo bimestre, es necesario tener claros los conceptos, técnicas de análisis y metodologías para proponer la construcción e implementación de una solución tecnológica. Si aún existen dudas por favor revise la bibliografía básica y complementaria o contacte con su docente/tutor a través de los distintos medios como correo electrónico, plataforma virtual (EVA) o a través de Zoom como servicio de videoconferencia basado en la nube.

Contenidos, recursos y actividades de aprendizaje



Semana 9 a 15

El objetivo de que haya cursado y desarrollado las actividades de los talleres planificados, es reforzar sus conocimientos teóricos y prácticos adquiridos en el transcurso de la carrera, de manera que sirvan como apoyo al momento de analizar, diseñar y proponer soluciones de forma documentada (textual y gráfica) de cara a su presentación al Examen Complexivo oral y Examen Complexivo escrito. Cumplido con este propósito procederemos a explicar con detalle cómo será el Examen Complexivo oral y el Examen Complexivo escrito.

Unidad 5. Evaluación del Examen Complexivo oral y escrito

Para presentarse a rendir el Examen Complexivo oral y el Examen Complexivo escrito es fundamental que usted conozca algunos tips, técnicas y buenas prácticas para el análisis, diseño, construcción, pruebas e implementación de una solución de software. Además, debe tener en cuenta las fases del proceso de gestión de proyectos y del ciclo de vida de desarrollo de software. Es importante que como metodologías de desarrollo conozca los marcos o frameworks como SCRUM, sus eventos, roles y artefactos. Finalmente, es requerido que para la construcción e implementación de soluciones de software haga uso de herramientas y tecnologías que le permitan aplicar buenas prácticas de desarrollo

de software empresarial. En este contexto, la presente unidad detalla el proceso de evaluación del Examen Complexivo tanto oral como escrito.

5.1. Examen Complexivo oral y escrito

El Examen Complexivo como opción de titulación de la Unidad de Integración Curricular comprende una evaluación escrita y una oral. Tanto la evaluación escrita como la evaluación oral se elabora con base en contextos o estudios de caso y su resolución contiene un rigor académico similar a la de un trabajo de titulación, de manera que, se evidencie por parte del estudiante el logro de las competencias específicas declaradas en el perfil de egreso de la carrera de Tecnologías de la Información.

Hasta el momento y con ayuda de las temáticas estudiadas en cada taller, usted debe estar en capacidad de presentarse a rendir el Examen Complexivo oral y el Examen Complexivo escrito.

Examen Complexivo escrito:

La metodología para la evaluación de las competencias, contemplará: Contextos o estudios de casos.

El Examen Complexivo escrito está diseñado con base en las temáticas abordadas en los talleres de las asignaturas Prácticum 4.1 y Prácticum 4.2, de los cuales, se derivarán un conjunto de preguntas objetivas de opción múltiple de respuesta única.

Para el desarrollo del Examen Complexivo escrito se debe tener presente los siguientes lineamientos:

- Todos los estudiantes que hayan aprobado la asignatura Prácticum 4.1 y desarrollado y aprobado todas las actividades planificadas en los talleres de la asignatura Prácticum 4.2, se presentarán a rendir el Examen Complexivo escrito.
- El Examen Complexivo escrito abarca los temas desarrollados en los talleres, contiene un conjunto de preguntas que el estudiante debe dar respuesta.
- El Examen Complexivo escrito puede desarrollarse de manera presencial o virtual, dependiendo de las disposiciones de la

institución y en las fechas determinadas para dicha actividad. Regularmente, se desarrolla junto con las evaluaciones del segundo bimestre de modalidad a distancia.

El Examen Complexivo escrito se califica sobre 10 puntos y representa el 35% de la calificación (3.5/10 puntos) de la asignatura Prácticum 4.2.

Examen Complexivo oral:

El Examen Complexivo oral está orientado a resolver casos o contextos que caracterizan o se asocian con problemas relacionados con las temáticas abordadas en los talleres de las asignaturas Prácticum 4.1 y Prácticum 4.2 y, además, sustentar los resultados frente a un tribunal, quienes evaluarán al estudiante con el debido rigor académico.

Para el desarrollo del Examen Complexivo oral se debe tener en cuenta los siguientes lineamientos:

- Todos los estudiantes que hayan aprobado la asignatura Prácticum 4.1 y desarrollado todas las actividades planificadas en los talleres de la asignatura Prácticum 4.2, se presentarán a rendir el Examen Complexivo oral.
- El Examen Complexivo oral puede desarrollarse de manera presencial o virtual, dependiendo de las disposiciones de la Universidad para dicha actividad. Regularmente, el Examen Complexivo oral se desarrolla a partir de la semana 14 del ciclo académico.
- El Examen Complexivo oral se divide en dos partes y se desarrollan en el mismo día:
 - Parte 1: Durante la jornada de la mañana, regularmente a partir de las 08H00 a.m., el estudiante desarrolla el examen práctico por un lapso de entre 4 y 5 horas ininterrumpidas. Durante este tiempo, el estudiante analiza, diseña y construye una solución a un problema planteado, el mismo que está alineado con las temáticas abordadas en los talleres del Prácticum 4.1 y Prácticum 4.2.
 - Parte 2: Durante la jornada de la tarde, regularmente a partir de las 14H00 p.m., se lleva a cabo la defensa oral del análisis, diseño, construcción e implementación de la solución al

problema resuelto en la jornada de la mañana. En esta actividad el estudiante por el lapso de una hora expone ante un tribunal conformado por docentes de la Carrera su propuesta de solución y brinda respuestas argumentadas y claras a las preguntas que el tribunal realice.

- Uno de los requisitos principales y fundamental para presentarse al examen complexivo (parte escrita y parte oral) es tener organizado y desarrollado los casos de estudio de los talleres, ya que estos se constituyen en la base para rendir el Examen Complexivo. Si el estudiante no tiene los casos de estudio desarrollados en los talleres no podrá presentarse a rendir el mismo.
- Todos los estudiantes deben elaborar una presentación usando herramientas como PowerPoint o Genially con la siguiente información: Problema a ser resuelto, contexto en alto nivel de su propuesta de solución, fundamentación teórica que la soporta, marcos de trabajo, estándares y herramientas utilizadas, fuentes bibliográficas utilizadas y exposición de la documentación que respalda su propuesta de solución (trabajo realizado en los talleres).

El Examen Complexivo oral se califica sobre 10 puntos y representa el 35% de la calificación (3.5/10 puntos) de la asignatura Prácticum 4.2.

Es importante aclarar que en la presente asignatura se obtienen dos calificaciones que son: la calificación del Examen Complexivo como opción de titulación y la calificación del Prácticum 4.2 como asignatura; las mismas que se detallan seguidamente:

- El Examen Complexivo como opción de titulación es calificado sobre 10 puntos, donde, el 50% de la nota se obtiene del Examen Complexivo oral y el otro 50% de la nota se obtiene del Examen Complexivo escrito. De la suma total de estas dos calificaciones con la respectiva ponderación debe dar como mínimo 7/10 puntos para aprobar Examen Complexivo como opción de titulación.
- El Prácticum 4.2 como asignatura es calificado sobre 10 puntos, donde, el 30% de la nota se obtiene de las actividades cumplidas en el primer bimestre, el 35% de la nota se obtiene del Examen Complexivo oral y el 35% de la nota del Examen Complexivo escrito. De la suma total de estas tres calificaciones con la respectiva ponderación debe

dar como mínimo 7/10 puntos para aprobar la asignatura Prácticum 4.2.

Recuerde que la calificación obtenida del Examen Complexivo oral más la calificación obtenida del Examen Complexivo escrito suman 10/10 puntos, de los cuales, como mínimo, debe obtener 7/10 puntos para aprobar la misma. En caso de no reunir la calificación mínima requerida, debe presentarse a la evaluación de recuperación (oral y escrita) en las fechas establecidas por la Universidad.

5.2. Rúbrica de Examen Complexivo oral

El Examen Complexivo oral se califica sobre 10 puntos y los parámetros se detallan seguidamente.

Tabla 2.

Rúbrica del examen complexivo oral

Parámetro	Criterio	Puntaje	DEFICIENTE 1%	REGULAR 25%	BUENO 50%	MUY BUENO 75%	EXCELENTE 100%
CASO PRÁCTICO DESARROLLADO EL DÍA DE LA DEFENSA	Entregables desarrollados el día de la defensa*	5	Desarrolla menos del 25% de entregables de acuerdo a los parámetros solicitados.	Desarrolla al menos el 25% de entregables de acuerdo a los parámetros solicitados.	Desarrolla al menos el 50% de entregables de acuerdo a los parámetros solicitados.	Desarrolla al menos el 75% de entregables de acuerdo a los parámetros solicitados.	Desarrolla el 100% de entregables de acuerdo a los parámetros solicitados.
DEFENSA	Argumentación y claridad de la exposición	3	La exposición no está relacionada con el trabajo realizado.	No expresa con claridad sus ideas y no utiliza los recursos de manera efectiva.	No expresa con claridad sus puntos de vista y utiliza los recursos de manera efectiva.	Expresa con claridad sus puntos de vista y no utiliza los recursos de manera efectiva.	Expresa con claridad sus puntos de vista y utiliza los recursos de manera efectiva.
	Respuestas	1	No responde a lo solicitado.	Las respuestas no son claras ni concisas.	Las respuestas son poco claras y concisas.	Las respuestas son claras y concisas.	Las respuestas son muy claras, concisas y demuestra buen manejo del tema.

Parámetro	Criterio	Puntaje	DEFICIENTE 1%	REGULAR 25%	BUENO 50%	MUY BUENO 75%	EXCELENTE 100%
	Proyección a futuro	1	En el trabajo desarrollado no hace uso de conceptos de tendencias tecnológicas actuales.	No encuentra aplicación futura del trabajo desarrollado.	En el trabajo desarrollado hace uso de conceptos de tendencias tecnológicas actuales, pero la exposición de los mismos son limitados.	Encuentra aplicabilidad de lo desarrollado en escenarios reales y usando conceptos de tendencias tecnológicas actuales.	Encuentra aplicabilidad de lo desarrollado en escenarios reales y propone aplicaciones futuras haciendo uso de conceptos de tendencias tecnológicas actuales.

* Los entregables específicos se colocarán de acuerdo al caso planteado y corresponderán como máximo a la nota indicada.

5.3. Ejemplo de caso de estudio a ser evaluado

El estudio de caso detallado en el anexo de la presente guía (Ver Anexo1, Anexo2, Anexo3) integran todas las temáticas abordadas en los talleres de la asignatura (taller "Arquitectura de Software", taller "Arquitectura Empresarial" y taller "Desarrollo de aplicaciones empresariales") y sirve como base para la evaluación del Examen Complexivo.



Actividad de aprendizaje recomendada

Realice una infografía donde resuma los conceptos utilizados dentro de cada uno de los talleres para reforzar la parte conceptual y práctica.



Semana 16



Actividades finales del bimestre

¡En hora buena! hemos finalizado el desarrollo y estudio de los conceptos básicos acorde a cada dominio académico, con los cuales usted ha sido

capaz de planificar, diseñar, construir e implementar una solución ante un escenario o problema real. Para apoyar a su comprensión y aprendizaje se han seleccionado e incluido una serie de conceptos básicos, se ha incluido una sección de anexos y se han cargado ejemplos en la plataforma educativa que sirvan de ayuda para que usted proponga una solución.

La práctica y experiencia que vaya adquiriendo o ya la tenga dentro del ámbito profesional donde se desenvuelve actualmente, le ayudará a decidir sobre el uso de herramientas, estándares o documentación base local o en la nube.

Durante este curso usted ha adoptado los roles de analista, diseñador, arquitecto de software, desarrollador y tester de calidad.

Me despido invitándole a que durante este tiempo pueda prepararse para rendir el examen complejivo parte oral y escrita donde se evaluará sus conocimientos y su experiencia para alcanzar el título de Ingeniero en Tecnologías de la Información. Le deseo muchos éxitos y la consecución de sus objetivos como parte de su formación profesional. ¡Hasta pronto!



4. Solucionario

Autoevaluación 1		
Pregunta	Repuesta	Retroalimentación
1	c	El principio de modularidad define que un sistema complejo puede dividirse en piezas más simples llamadas módulos y que un sistema compuesto de módulos es llamado modular.
2	a	El principio de abstracción es un proceso mediante el cual se identifican los aspectos relevantes de un problema ignorando los detalles; es un caso especial del principio de separación de intereses, en el cual se separan los aspectos importantes de los detalles de menor importancia.
3	b	Las metodologías ágiles siguen los modelos de procesos iterativos e incrementales. Son iterativos en el sentido de que planifican que el trabajo de una iteración mejore. En iteraciones posteriores. Son incrementales porque el trabajo terminado se entrega a lo largo del proyecto.
4	a	Scrum es una metodología ágil flexible constituida por un conjunto de buenas prácticas que permiten a los equipos trabajar colaborativamente en entregables pequeños y priorizados para brindar el máximo valor posible a los clientes desde las primeras entregas.
5	a	Scrum es una metodología ágil flexible constituida por un conjunto de buenas prácticas que permiten a los equipos trabajar colaborativamente en entregables pequeños y priorizados para brindar el máximo valor posible a los clientes desde las primeras entregas.
6	a	DevOps es el vínculo e integración de los equipos de desarrollo y operaciones para abordar un proyecto de <i>software</i> .
7	a	La integración continua es una práctica de desarrollo de <i>software</i> donde todos los miembros del equipo integran el código con frecuencia y este es verificado de forma automática a través de un servidor de integración.
8	c	La entrega continua es la extensión de la integración continua, su objetivo es recuperar el código fuente, compilarlo y desplegarlo para verificar el funcionamiento de la aplicación.
9	a	Una canalización garantiza la entrega de <i>software</i> rápido y de calidad.

Autoevaluación 1

Pregunta Repuesta Retroalimentación

10

a

Una arquitectura de micro-servicios consiste en una colección de pequeños servicios autónomos. Cada servicio es autónomo y debe implementar una capacidad comercial única dentro de un contexto delimitado. Un contexto acotado es una división natural dentro de un negocio y proporciona un límite explícito dentro del cual existe un modelo de dominio.

Ir a la
autoevaluación

Autoevaluación 2

Pregunta	Respuesta
1	a
2	c
3	c
4	a
5	a
6	a
7	c
8	a
9	b
10	c

Ir a la
autoevaluación

Autoevaluación 3		
Pregunta	Respuesta	Retroalimentación
1	a, b, c	A nivel de programación cuando interactuamos con una base de datos podemos realizar las operaciones básicas, así como programar excepciones que permitan presentar mensajes personalizados a los usuarios finales.
2	c	<p>La sentencia INSERT INTO se utiliza para insertar nuevos registros en una tabla. Es posible escribir la declaración INSERT INTO de dos maneras:</p> <ol style="list-style-type: none"> 1. Especifique tanto los nombres de las columnas como los valores que se insertarán. 2. Si agrega valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, asegúrese de que el orden de los valores sea el mismo que el de las columnas de la tabla.
3	a	Cuando se trabaja con genéricos, en cualquier lenguaje de programación, se debe declarar los tipos de datos iguales, tanto en el lado izquierdo como en el derecho.
4	a	<p>La sentencia INSERT INTO se utiliza para insertar nuevos registros en una tabla. Es posible escribir la declaración INSERT INTO de dos maneras:</p> <ol style="list-style-type: none"> 1. Especifique tanto los nombres de las columnas como los valores que se insertarán. 2. Si agrega valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, asegúrese de que el orden de los valores sea el mismo que el de las columnas de la tabla.
5	a	La declaración DELETE se usa para eliminar registros existentes en una tabla.
6	d	SQL es un lenguaje estándar para acceder y manipular bases de datos.
7	c	Java.sql proporciona la API para acceder y procesar datos almacenados en una fuente de datos (generalmente una base de datos relacional), utilizando el lenguaje de programación Java.
8	a	<p>La sentencia INSERT INTO se utiliza para insertar nuevos registros en una tabla. Es posible escribir la declaración INSERT INTO de dos maneras:</p> <ol style="list-style-type: none"> 1. Especifique tanto los nombres de las columnas como los valores que se insertarán. 2. Si agrega valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, asegúrese de que el orden de los valores sea el mismo que el de las columnas de la tabla.

Autoevaluación 3

Pregunta	Respuesta	Retroalimentación
9	b	<p>La sentencia INSERT INTO se utiliza para insertar nuevos registros en una tabla. Es posible escribir la declaración INSERT INTO de dos maneras:</p> <ol style="list-style-type: none">1. Especifique tanto los nombres de las columnas como los valores que se insertarán.2. Si agrega valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, asegúrese de que el orden de los valores sea el mismo que el de las columnas de la tabla.
10	a	<p>La clase <i>statement</i> dispone de un método llamado executeUpdate el cual recibe como parámetro la cadena de caracteres que contiene la sentencia SQL a ejecutar. Este método únicamente permite realizar sentencias de actualización de la BD: creación de tablas (CREATE), inserción (INSERT) y borrado de datos (DELETE). En este caso el valor que retorna es un valor entero.</p>

[Ir a la
autoevaluación](#)



5. Referencias bibliográficas

5.1. Bibliografía Básica

Guamán, D. Benítez, S. (2022). *Guía Didáctica de Examen Complexivo Prácticum 4.2*. Loja, Ecuador: Ediloja.

El propósito de esta guía es presentar una recopilación de temas que han sido previamente cursados por los estudiantes durante la carrera universitaria y además, guiar a los estudiantes en el Examen Complexivo oral y escrito.

5.2. Bibliografía Complementaria

CES (2020). Reglamento de régimen académico. Reformado mediante RPC-SO-08-No.111-2019.

Abad, P. (2019). Guía Didáctica de Fundamentos de Ingeniería de Software. Loja, Ecuador: Ediloja.

La guía didáctica se ha elaborado para orientar al estudiante en su aprendizaje, contiene todos los elementos necesarios que le permita estudiar de forma autónoma.

Guamán, D., (2021). Arquitectura de Software. Loja, Ecuador: UTPL

El propósito de esta guía es presentar una recopilación de temas que involucran conceptos de Arquitectura de Software. La información presentada guiará a los estudiantes en la revisión esquematizada de los contenidos y complementará la bibliografía proporcionada.

Sucunuta, M. (2017). Guía Didáctica de Ingeniería de Requisitos. Loja, Ecuador: Ediloja

Texto Guía elaborado para el estudio de la asignatura de Ingeniería de Requisitos en la titulación de Ingeniería en Informática de la Modalidad Abierta y a Distancia de la Universidad Técnica Particular de Loja. Se cubre el proceso de desarrollo de requisitos: Obtención,

análisis, especificación y validación; así como también aquellas actividades de gestión de requisitos. En cada uno de los temas que se abordan se proponen cuestionarios y ejemplos que permiten la adecuada comprensión por parte del estudiante. Contiene todos los elementos necesarios que le permita estudiar de forma autónoma.

Cueva, S.; Rodríguez, G. (2020). Gestión de la calidad del software. Loja, Ecuador: Ediloja.

Guía didáctica virtualizada elaborada para la asignatura de Gestión de la calidad de software de la modalidad de estudios a distancia de la UTPL, carrera de Tecnologías de la Información. En la cual se abarcan temas relevantes relacionados a la gestión de la calidad; calidad de personas, servicios, productos, proyecto y los principales estándares y modelos que sirven para implementar calidad en los diferentes ámbitos del ciclo de vida de desarrollo de un producto software.

Torres, Y., & Cabrera, A. (2021). Arquitectura Empresarial Loja: EdiLoja.

El texto guía le muestra los lineamientos necesarios para que usted conozca cómo puede representar arquitectónicamente a una organización de manera integral, cubriendo todos los elementos que la conforman y sus relaciones, a través de dominios de negocio, información, aplicaciones y tecnología.

Guamán, D., (2018). Modelado. Loja, Ecuador: UTPL

El propósito de esta guía es presentar una recopilación de temas que involucran aspectos de Modelado. La información presentada guiará a los estudiantes en la revisión esquematizada de los contenidos y complementará la bibliografía proporcionada.

Soto, F. y Correa, R. (2020). Texto Guía de Metodologías de Desarrollo. Loja, Ecuador: Ediloja.

El Texto Guía de Metodologías de desarrollo le permitirá conocer cómo manejar un proceso de desarrollo de software mediante la aplicación de modelos de procesos de software, metodologías y frameworks tradicionales o ágiles, y la automatización de las fases del ciclo de desarrollo de software.

Morocho, J. C., Encalada, E., & Romero, A. (2020). Administración de Base de Datos. Guía Didáctica (1a. ed.). Loja, Ecuador: UTPL.

Instrumento didáctico que se centra en los temas que son objeto de estudio en esta asignatura. Direcciona hacia los contenidos que se deben revisar tanto en el texto básico como en otros recursos fundamentales. Complementariamente ofrece orientaciones, ejemplos, y actividades, que permiten aclarar, profundizar, aplicar y evaluar lo aprendido.

Torres, D. (2019). Guía didáctica Tecnologías Web. Loja: Editorial UTPL.

Esta guía es un instrumento didáctico que se usará como complemento del texto básico, en el cual usted encontrará lineamientos que lo guiarán a través del texto básico; además de brindar orientaciones adecuadas para facilitar el proceso de enseñanza-aprendizaje de esta asignatura.

Abad, M. (2021) Guía didáctica: Gestión de Proyectos. Ediloja. Loja-Ecuador.

La guía didáctica describe los principales procesos de gestión de proyectos orientados al campo de Tecnologías de la Información, utiliza como recursos al texto básico el cual es complementado con información actualizada de diferentes fuentes y de la experiencia del autor. Desarrolla un caso práctico que ayuda en la comprensión de la temática y favorece el desarrollo de las competencias.

Cabrera, A., & Torres, Y. (2020). Planificación estratégica y Sistemas de Calidad Empresarial. Loja: EdiLoja.

El texto guía le muestra métodos efectivos de planificación estratégica en donde se establecen rutas, procesos, planes y recursos que puede utilizar para llevar a cabo un método de planificación estratégica y así poder impulsar el entorno digital en la organización.

Cueva, S.; Picoita, Galo.; (2021). Texto Guía de Organización y Administración de Infraestructura de TI. Loja- Ecuador. UTPL.

El texto guía le ayudará a enfocarse en el estudio de la asignatura, además de organizar el tiempo dedicado a cada una de las unidades.

5.3. Recursos Educativos Abiertos (REA's)

Título del REA	URL de acceso
OCW1_OpenCourseWare, <i>Universidad de Cantabria</i>	https://ocw.unican.es/course/view.php?id=169&section=2
OCW2_OpenCourseWare, <i>Universidad del País Vasco / Euskal Herriko Unibertsitatea</i>	https://ocw.ehu.eus/pluginfile.php/51913/mod_resource/content/6/Pruebas-OCW.pdf
OCW3_OpenCourseWare, <i>Universitat de Lleida</i>	http://ocw.udl.cat/enginyeria-i-arquitectura/enginyeria-del-software-iii/Continguts/1%20-%20Introduccio%20-Arquitectura.pdf
OCW4_OpenCourseWare, <i>Universidad Carlos III de Madrid</i>	http://ocw.uc3m.es/ingenieria-informatica/sistemas-distribuidos-2013/Tema8_serviciosweb.pdf/view http://ocw.uc3m.es/ingenieria-informatica/sistemas-distribuidos-2013/Tema4_aplicacionesclienteservidor.pdf/view
Documentos Digitales	Descripción y nombre de archivo
DD1_Arquitecturas software, Departamento de Ciencias de la Computación e I.A, Universidad de Granada, documento digital	Resumen 4+1, Estilos y Patrones.pdf (Documento digital que se cargará en la plataforma CANVAS, contiene temas: 4+1 vistas, Patrones arquitectónicos, Proceso de diseño de arquitectura, Validación de la arquitectura)
DD2_Arquitectura de aplicaciones	Diseño de aplicaciones en capas.pdf (Documento digital que se cargará en la plataforma CANVAS, contiene temas: explicación con diagramas y código de tiers, layers, DAO, DTO)
DD3_Arquitecturas de Software, Guía de estudio	Guía Arquitectura de Software.pdf (Documento digital que se cargará en la plataforma CANVAS, contiene temas: calidad del software, Importancia de la arquitectura de software, Componentes, conectores y relaciones, Calidad arquitectónica, Atributos de calidad, ISO/IEC 9126 adaptado para arquitecturas de software, Relación entre Arquitectura de Software y Atributos de Calidad, Estilo Arquitectónico, Patrón Arquitectónico, Vistas Arquitectónicas, Evaluación de arquitecturas de software)
Lecturas recomendadas en la Web	Descripción y URL de acceso

Título del REA	URL de acceso
Lecturas recomendadas en la Web: Evolución de las arquitecturas de Software	Arquitecturas monolitica vs microservicios https://openwebinars.net/blog/diferencia-entre-arquitectura-monolitica-y-microservicios/ Conversión de aplicaciones monolíticas en microservicios mediante el diseño basado en dominios https://docs.microsoft.com/es-es/azure/architecture/microservices/migrate-monolith De una arquitectura tradicional a microservicios https://www.hiberus.com/crecemos-contigo/de-una-arquitectura-tradicional-a-microservicios/
Lecturas recomendadas en la Web: Estilos vs Patrones Arquitectónicos	Estilos y patrones arquitectónicos Architectural Styles vs. Architectural Patterns vs. Design Patterns https://bit.ly/2NVxrq3
Lecturas recomendadas en la Web: El modelo C4 de documentación para la Arquitectura de Software	https://bit.ly/3pQoV9R
Lecturas recomendadas en la Web: Todo lo que necesitas saber sobre la arquitectura de aplicaciones móviles	https://www.peerbits.com/blog/all-about-app-architecture-for-efficient-mobile-app-development.html
Documentos y plantillas	URL de descarga
Plantillas para documentación arquitectónica	P1: http://www.se.rit.edu/~co-operators/SoftwareArchitectureDocumentation.pdf P2: https://projects.cecs.pdx.edu/attachments/download/3707/Software_Architecture_Document.docx

Nota: Los documentos digitales etiquetados como DD y las plantillas para documentación arquitectónica se cargarán a la plataforma7



6. Anexos

Temáticas generales, herramientas y estándares

Dominio	Insumos	Entregables Comunes (Talleres)	Temáticas Generales	Herramientas / Estándares
Arquitectura	Caso General Listado de requerimientos funcionales / no funcionales	E01: Diagrama de Contexto E02: Lista de procesos E03: Lista de necesidades generales E04: Documento de visión E05: Registro de interesados E06: Requisitos de usuario, requisitos funcionales, requisitos no funcionales E07: Acta de constitución del proyecto E08: Documento de especificación de requisitos Proceso SCRUM: <ul style="list-style-type: none">• Backlog de producto (Temas, Épicas, Historias, Tareas) 4 +1 View (Krutch) <ul style="list-style-type: none">• Escenarios• Casos de Uso• Clases• Componentes• Despliegue Arquitectura Lógica (MVC) y Arquitectura Física (Tiers)	Ingeniería de Software (SDLC + SDLCS) Ingeniería de requisitos (Requisitos de Negocio, Usuario, Sistema) Metodología de desarrollo Ágil (Backlog de producto) Arquitectura de Software (Definición y diseño de una arquitectura lógica y física) Arquitecturas de Software Evolución	Laboratorio Virtual: Enterprise Architect – UML
Desarrollo de Aplicaciones	Caso General Arquitectura de solución (MVC) On- premise y Cloud Modelo Lógico de datos Diagrama de componentes Diagrama de UI	Aplicación (Front – end + Back – end) Modelo físico BDD Gestión de código Git Plan de pruebas (Funcionales) SonarQube (Calidad y Seguridad)	Front end Back end GitHub comandos básicos	Front – end (HTML, CSS3) Cualquier tecnología

Dominio	Insumos	Entregables Comunes (Talleres)	Temáticas Generales	Herramientas / Estándares
Arquitectura Empresarial	Caso General Arquitectura de Solución	Mapa de capacidades empresariales Modelo Lógico de datos Diagrama de componentes Diagramas de UI (Figma)	Puntos de Vista Dominios de arquitectura empresarial Arquitectura de negocio Arquitectura de TI Transición	Laboratorio Virtual: Enterprise Architect – UML

ANEXO 1: Escenario o estudio de caso

En vista que el perfil profesional del Ingeniero en TI, lo habilita para dar soluciones empresariales, lo que se le solicita a usted es **“Diseñar y proponer una solución documentada para el desarrollo de un Sistema de Gestión Organizacional para PYMES-Ecuador (ARQSOFT)”**.

Las necesidades se han redactado de forma tal que Ud. pueda analizarlas y con ellas proveer de una solución. Antes de ello, es necesario que tenga en cuenta lo siguiente:

NECESIDADES: Luego de una reunión entre los involucrados del proyecto, se recaban y listan las siguientes necesidades las cuales servirán para determinar alcance, tiempo, costos, recursos, riesgos entre otros.

- Que exista un módulo a nivel de sistema que permita la gestión de roles, permisos y usuarios (RPU).
- Que existan módulos y funcionalidades a nivel de sistema que permitan la gestión de convocatorias, directiva, sesiones, socios, compras, ventas y los que considere para ARQSOFT.
- Que el módulo de RPU permita la integración e interoperabilidad con el resto de los módulos y componentes de la solución que Ud. proponga.
- Que el módulo de RPU sea el CORE (similar a un sistema de seguridad central) para gestionar el acceso a todos los componentes y módulos que forman parte de ARQSOFT.
- Que se tenga en cuenta aspectos de confidencialidad para los datos e información considerados sensibles, los cuales son intercambiados entre un emisor y uno o varios receptores (cliente(s), sistema(s), base de datos).
- Que las personas que hacen uso de ARQSOFT puedan trabajar sin inconvenientes en los horarios de 08h00 a 18h30.
- Que cada 30 días en el horario de 19h00 de forma automática se generen respaldos de los datos almacenados en la base de datos local de ARQSOFT y se almacenen en un repositorio de Cloud.

- Los datos que son parte de las operaciones sobre el sistema deben ser alojados en un repositorio local y en la nube.
- Que las operaciones de consulta de datos que se realizan desde la interfaz gráfica de cada funcionalidad no superen los 30 segundos.
- Que las operaciones de edición de datos que se realizan desde la interfaz gráfica de cada funcionalidad no superen los 5 segundos.
- Tanto el administrador como los usuarios del sistema pueden acceder al sistema desde cualquier lugar a los módulos y funcionalidades asociados a ellos.
- Todos los módulos deben tener opciones para realizar las operaciones CRUD.
- Todos los módulos deben tener funcionalidades para emitir o generar reportes que antes de ser impresos deben ser visualizados.
- Reportes como Convocatorias, Sesiones, Directiva, Socios, Compras y Ventas deben tener un espacio para la firma y marca de agua en la parte central de la hoja que se imprima.
- Que los reportes que generan las funcionalidades del sistema se puedan exportar en formato pdf, xls, xml, csv.
- ARQSOFT debe operar adecuadamente con 100 usuarios concurrentes.
- Las comunicaciones externas entre servidor(es) de datos, aplicación y cliente(s) del sistema deben hacer uso de protocolos como HTTPS o HTTSPS.
- Los datos considerados confidenciales deben estar encriptados, haciendo uso de algoritmos de encriptación y desencriptación.
- Las compras y ventas de productos deberán ser registrados a través de una interfaz que permita interoperabilidad con los servicios del Banco XYZ.

- Para constancia de que se ha realizado una compra o venta de uno o varios productos, el sistema deberá generar una factura la cual será electrónica, para ello usará servicios provistos por el SRI o de proveedores para este fin.
- Para la construcción de los nuevos módulos de ARQSOFT se debe seguir utilizando como entorno de desarrollo Microsoft Visual Studio.
- Que la documentación derivada del proyecto y producto se gestione en un repositorio el cual está alojado en Cloud.
- Por cada acción importante que se realice sobre el sistema se debe considerar el almacenamiento de datos en archivos de log's o blogs en tablas de un modelo de datos, estos datos deberán contener al menos fecha, hora, módulo, funcionalidad, acción.
- Todos los usuarios que tengan acceso al sistema a través de un rol podrán hacer uso del módulo de gestión de reportes. Este módulo deberá usted proponer la forma de realizarlo para satisfacer las necesidades.

Para conocer a detalle y clasificar lo que se requiere como parte de la Información Organizacional y Gestión Financiera donde intervienen módulos como RPU, Convocatorios, Sesiones, Socios, Compras y Ventas se expone lo siguiente.

Gestión de roles, permisos y usuarios (RPU)

La gestión implica la realización de operaciones CRUD (Create, Read, Update, Delete) que, a través de la interacción entre el cliente, sistema(s), base de datos y/o servicios permitirá:

- Crear, editar, eliminar, listar los usuarios.
- Crear, editar, eliminar, listar los módulos.
 - Los módulos que forman parte de ARQSOFT son:
 - Gestión de Convocatorias
 - Gestión de Sesiones
 - Gestión de Socios
- Crear, editar, eliminar, listar los roles.
 - Los tipos de roles que soporte el sistema serán: Administrador, Técnico1, Técnico2, Técnico 3, Auditor.

- Rol Administrador: tendrá permisos para visualizar y operar sobre todos los módulos y funcionalidades del sistema.
- Rol Técnico1: tendrá permisos para visualizar y operar sobre los módulos y funcionalidades:
 - Gestión de Socios
 - Gestión de Convocatorias
 - Gestión de Sesiones
- Rol Técnico2: tendrá permisos para visualizar y operar sobre el módulo y funcionalidades de:
 - Gestión de Socios
- Rol Auditor: tendrá permisos para visualizar y operar sobre el módulo y funcionalidades de:
 - Gestión de Sesiones

A través de RPU se podrá entre otras cosas:

- Asignar y desasignar módulos a roles
- Asignar y desasignar roles a usuarios
- Generar reportes de roles, permisos y usuarios
- Generar logs de actividades de asignación/desasignación
- Debe analizar todos los casos posibles, por ejemplo, considerar que a un usuario se le puede configurar uno o varios roles y que dicho usuario a través de su rol podrá tener acceso a uno o varios módulos que forman parte del sistema y que dicho módulo tendrá una o varias funcionalidades.
- Además, considere algún algoritmo o tecnología para la generación de credenciales y para acceso al sistema a través de dichas credenciales.
- Considere utilizar para la autenticación de usuarios servicios externos como por ejemplo OAuth y servicios provistos por el Registro Civil del Ecuador (<https://www.registrocivil.gob.ec/web-service-2/>)

Gestión de Convocatorias

Como parte de la *Información Organizacional* de cada PYME, uno de los procesos necesarios es la Gestión de Convocatorias. Esto implica que a nivel de sistema exista un módulo que a través de operaciones CRUD permita realizar los procesos de registro, edición, consulta y eliminación de Convocatorias. Para proceder a crear una Convocatoria es necesario tener previamente configurado los Socios y Directiva a través de los módulos y funcionalidades respectivas, de no tenerlos se debe realizar primero este proceso.

Para registrar una Convocatoria se solicitará datos como el tipo de organización el cual puede ser Cooperativa, Asociación, Organismo Comunitario. Dependiendo de la selección acorde a la Organización se desplegará los tipos de asamblea los cuales pueden ser Asamblea General, Consejo de administración y Consejo de Vigilancia. El tipo de Sesión puede ser Ordinaria, Extraordinaria e Informativa y finalmente para registrar la convocatoria se requiere especificar el lugar, fecha de convocatoria y hora de registro de la convocatoria. Considere que una convocatoria es creada para llevar a cabo una sola sesión.

Gestión de Directiva

En el módulo de Gestión de Directiva se puede llevar a cabo la configuración de la Directiva que regirá por un periodo de tiempo de 1 año calendario. Para la elección de presidente, secretario y vocales que forman parte de la Directiva, se debe elegir entre los Socios que han sido previamente configurados en el sistema en el módulo de Socios. Al ser un módulo de gestión recuerde lo que implica proponer para el correcto funcionamiento, es decir que se pueda llevar a cabo el registro, edición, consulta, eliminación.

Para elegir la Directiva debe existir una Convocatoria de Socios. El proceso de elección inicia verificando que exista el quórum respectivo (más del 50% de socios) para esto se utiliza una funcionalidad del sistema que permite tomar y registrar asistencia de los Socios a la Convocatoria. Una vez que se ha comprobado la existencia de quórum, de todos los Socios registrados en el sistema se propone uno o varios, luego de ello todos los Socios que físicamente se encuentran en la fecha y hora de la Convocatoria proceden a registrar su voto de forma anónima y en papeles para depositarlos en un ánfora donde de luego de forma manual se realizará el conteo. El Socio que

obtenga el mayor número de votaciones será el presidente y el que le sigue en votos será el secretario. Estos dos actores son los principales dentro de la Organización, luego ellos son los encargados de seleccionar a los demás miembros de la Directiva. (*El proceso de votación y conteo de votos podría automatizarse como parte de la solución*)

Gestión de Sesiones

La Sesión es parte de una Convocatoria, algunas de las reglas o requisitos de esta son los siguientes:

- Para llevar a cabo una sesión debe existir una convocatoria asociada.
- La sesión deberá iniciar en la fecha y hora indicada de la convocatoria.
- El secretario o secretaria, a través de su dispositivo móvil o un equipo portátil debe utilizar la funcionalidad del sistema que permite tomar lista de los asistentes a la sesión.
- La funcionalidad de registrar asistencia permite visualizar todos los socios y marcar su asistencia o inasistencia.
- Debe existir como quórum máximo el 50% de los socios para iniciar la sesión.
- El presidente y secretario que son parte de la Directiva deben estar presentes como parte del quórum.
- En caso de que no exista quórum el estado de la sesión será postergado, por tanto, se debe editar su fecha de convocatoria y sesión y finalmente se imprimirá un acta con el estado y observaciones de la sesión.
- En caso de que se instale la sesión, el secretario debe tomar nota de la sesión a través de la funcionalidad del sistema. En esta funcionalidad se puede añadir uno o varios ítems (líneas) y puntos tratados (líneas).
- Al finalizar la sesión se debe imprimir el documento que contenga los datos de la convocatoria y sesión donde consten los ítems, acuerdos o puntos tratados en la sesión.

- Recuerde que al ser un documento formal deberá visualizarse para imprimirse para ser firmado al final de las hojas.
- Los datos de cada sesión (actual o anterior) pueden ser recuperados para realizar actividades de visualización o edición.

Gestión de Socios

Los socios se consideran como un conjunto de personas que forman parte de una PYME. Por tanto, a través de este módulo se puede:

- Registrar, editar, eliminar, listar Socios.
 - Identificación, Nombres, Género, Fecha de Nacimiento, Fecha de ingreso, Provincia, Cantón, Ciudad y Parroquia de residencia.
- Generar reportes de socios usando diferentes filtros o criterios de búsqueda.

ANEXO 2: Documento de Arquitectura de Software

Software Architecture Document

(Documento de Arquitectura de Software)

Proyecto ARQSOFT

Elaborado por: Daniel Alejandro Guamán Coronel

Versión 1.0

12/03/2021

Historial de revisiones

Versión	Descripción de versiones / cambios	Responsable	Fecha
1.0	Versión inicial	Daniel Alejandro Guamán Coronel	12/03/2021

Bloque de aprobación

Versión	Comentarios	Responsable	Fecha
1.0	En revisión		12/03/2021

Tabla de contenidos

1. Introducción
 - 1.1. Propósito
 - 1.2. Ámbito
 - 1.3. Definiciones, Acrónimos, y Abreviaturas
 - 1.4. Referencias
 - 1.5. Resumen
2. Representación arquitectónica
3. Objetivos y restricciones arquitectónicas
4. Vista de casos de uso
 - 4.1. Actores
 - 4.2. Realización de casos de uso
 - 4.2.1. Caso de uso Inicio de Sesión
 - 4.2.2. Caso de uso Solicitar Reporte
 - 4.2.3. Caso de uso Gestionar Usuarios
5. Vista de Procesos
 - 5.1. Resumen
 - 5.2. Realización de diagrama de secuencia
 - 5.2.1. Diagrama de secuencia inicio de sesión usando el servicio OAuth
 - 5.2.2. Diagrama de secuencia Gestionar Usuarios (Registrar un Usuario en la Base de Datos)
6. Vista Lógica
 - 6.1. Resumen
 - 6.2. Realización de diagrama de Clases
7. Vista de desarrollo
8. Vista de implementación

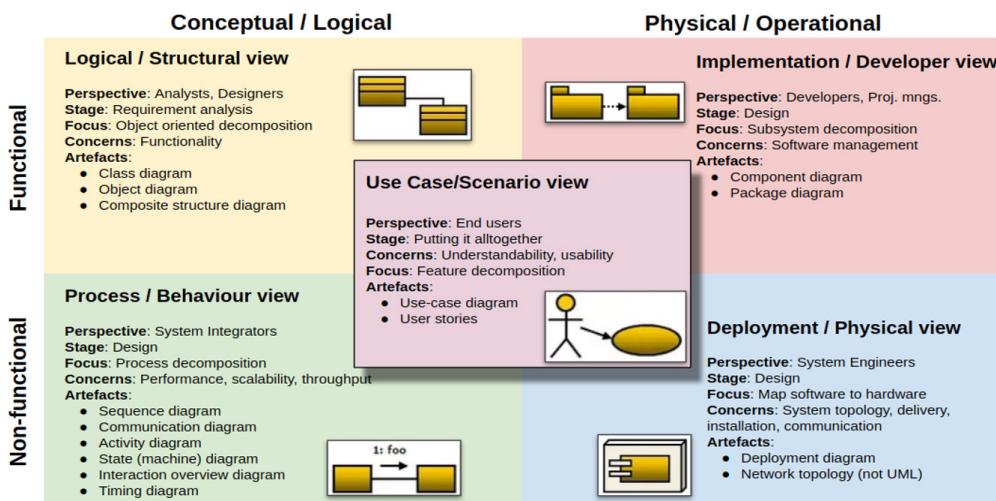
Software Architecture Document

1. Introducción

El presente documento proporciona una descripción general de alto nivel y explica la arquitectura del sistema ARQSOFT. El documento define los objetivos de la arquitectura, los casos de uso admitidos por el sistema, los estilos arquitectónicos y los componentes seleccionados. El documento proporciona una justificación para las decisiones de arquitectura y diseño tomadas desde la idea conceptual hasta su implementación.

1.1. Propósito

El Software Architecture Document (SAD, por sus siglas en inglés) proporciona una descripción general completa de la arquitectura del sistema ARQSOFT. El documento presenta una serie de vistas arquitectónicas para representar los diferentes aspectos del sistema. La estructura de este documento se basa en el modelo de arquitectura "4 + 1" propuesto por Philippe Kruchten [Kruchten].



El modelo "4 + 1" vistas permite a las partes interesadas apoyarse en la documentación para entender y comprender la arquitectura del software.

1.2. Ámbito

El ámbito de SAD es explicar la arquitectura del sistema ARQSOFT. Este documento describe los diversos aspectos de diseño que se consideran arquitectónicamente significativos. Estos elementos y comportamientos

son fundamentales para orientar la construcción del sistema ARQSOFT y para entender el proyecto en su conjunto. Se alienta a las partes interesadas que requieran un conocimiento técnico del sistema a iniciar primeramente con la lectura de la documentación de la propuesta de proyecto y los documentos de especificación de requisitos de software [SRS] desarrollados para este sistema.

1.3. Definiciones, Acrónimos, y Abreviaturas

- **HTTP** – Hypertext Transfer Protocol
- **WWW** – World Wide Web
- **GUI** – Graphical User Interface
- **SAD** - Software Architecture Document
- **SRS** – Software Requirements Specification
- **UML** – Unified Modeling Language
- **MVC** – Siglas del Patrón arquitectónico conformado por 3 partes que son: Modelo, Vista y Controlador
- **Modelo** – Es la capa que interactúa con los datos.
- **Vista** – Contienen el código que permite visualizar los resultados en la GUI.
- **Controlador** – Contiene el código para responder a las peticiones que realiza el cliente.
- **Laravel** – Framework de código abierto para desarrollar aplicaciones y servicios web con PHP y MVC.
- **MySQL** – Es un sistema de gestión de bases de datos relacional.

1.4. Referencias

[ERS]: Especificación de Requisitos de Software (SRS, por sus siglas en inglés)

[CRC]: Tarjetas CRC (Clase, Responsabilidad, Colaboración)

[Kruchten]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

1.5. Resumen

Para documentar todos los aspectos de la arquitectura, el Documento de Arquitectura de Software (SAD) contiene las siguientes subsecciones.

Sección 2: describe el uso de cada vista.

Sección 3: describe los objetivos arquitectónicos y las restricciones del sistema.

Sección 4: describe las realizaciones de los casos de uso principales.

Sección 5: describe la vista de procesos principales del sistema.

Sección 6: describe la vista lógica del sistema, incluidas las definiciones de interfaz y operación.

Sección 7: describe la vista de desarrollo que permitirá la construcción del sistema.

Sección 8: describe cómo el sistema se implementa a través del diagrama de despliegue.

2. Representación arquitectónica

Este documento detalla la arquitectura utilizando las vistas definidas en el modelo "4 + 1" [Kruchten]. Las vistas utilizadas para documentar el sistema ARQSOFT son:

2.1. Vista de casos de uso

Audiencia: todas las partes interesadas del sistema, incluidos los usuarios finales.

Área: describe el conjunto de escenarios y / o casos de uso que representan alguna funcionalidad central significativa del sistema. Describe los actores y los casos de uso del sistema, esta vista presenta las necesidades del usuario y se elabora más en el nivel de diseño para describir los flujos discretos y las restricciones con más detalle. Este vocabulario de dominio es independiente de cualquier modelo de procesamiento o sintaxis de representación.

Artefactos relacionados: diagrama de caso de uso, documentos de especificación de caso de uso.

2.2. Vista de procesos

Audiencia: Integradores de sistemas.

Área: *Requisitos no funcionales*: se ocupa de los aspectos dinámicos del sistema, explica los procesos del sistema y cómo se comunican, y se centra en el comportamiento del sistema en tiempo de ejecución.

Artefactos relacionados: diagrama de actividades, diagrama de secuencia.

2.3. Vista Lógica

Audiencia: Diseñadores.

Área: *Requisitos funcionales*: describe el modelo de objeto del diseño. También describe las realizaciones de casos de uso más importantes y los requisitos de negocio del sistema.

Artefactos relacionados: diagrama de clases, diagrama de objetos.

2.4. Vista de implementación/Desarrollo

Audiencia: Desarrolladores.

Área: *Requisitos funcionales*: implementa el modelo de objeto del diseño. Implementa a nivel de sistema los requisitos de negocio.

Artefactos relacionados: diagrama de componentes, diagrama de paquetes.

2.5. Vista de Despliegue

Audiencia: gerentes de implementación.

Área: **Topología**: describe la asignación del software al hardware y muestra los aspectos distribuidos del sistema. Describe las estructuras de implementación potenciales, al incluir escenarios de implementación conocidos y anticipados en la arquitectura, permitimos a los implementadores hacer ciertas suposiciones sobre el rendimiento de la red, la interacción del sistema, etc.

Artefactos relacionados: modelo de implementación.

2.6. Vista de Datos (*requerida cuando se trabaja con repositorio de datos relacional – no relacional*)

Audiencia: especialistas en datos, administradores de bases de datos

Área: Persistencia: describe los elementos persistentes arquitectónicamente significativos en el modelo de datos, así como cómo fluyen los datos a través del sistema.

Artefactos relacionados: modelo de datos.

3. Objetivos y restricciones arquitectónicas

Existen algunos requisitos clave y limitaciones del sistema que tienen una influencia significativa en la arquitectura, entre los que destacan:

1. El sistema está diseñado como una prueba de concepto para un sistema de gestión de datos e información principalmente para las Pymes del Ecuador que debe ser desplegado en la Web, sin embargo, puede escalar para adicionar nuevos módulos, funcionalidades o servicios que se construirán en el futuro. Por lo tanto, uno de los principales interesados en este documento y el sistema en su conjunto son los actuales y futuros arquitectos y diseñadores, no necesariamente los usuarios como suele ser el caso. Como resultado, uno de los objetivos de este documento es que pueda ser útil para los arquitectos, diseñadores y partes interesadas.
2. El sistema debe permitir comunicarse con varias API de sistemas, subsistemas o servicios de terceros. En este caso usaremos OAuth y los servicios expuestos por el Registro Civil para validar datos (<https://www.registrocivil.gob.ec/web-service-2/>). Definir cómo el sistema interactúa con estos sistemas de terceros es una preocupación principal de la arquitectura.
3. El sistema se construirá utilizando lenguajes de programación Java o PHP, y frameworks como Spring Tool Suite o Laravel. utilizará un sistema RDBMS (Sistema de gestión de base de datos relacional) de código abierto (MySQL) para la persistencia de datos y se implementará en un servidor web Linux. Estos requisitos especiales de implementación requieren una consideración adicional en el desarrollo de la arquitectura.
4. La Especificación de Requisitos de Software [ERS], describe una serie de cambios anticipados que la aplicación podría

enfrentar con el tiempo. Uno de los objetivos principales de la arquitectura del sistema es minimizar el impacto de estos cambios disminuyendo la cantidad de código que se necesitaría modificar para implementarlos. La arquitectura busca hacer esto mediante el uso de principios de diseño como la modularización, alta cohesión, bajo acoplamiento y ocultación de información para aislar los componentes que probablemente cambiarán del resto del sistema.

4. Vista de casos de uso

El propósito de la vista de casos de uso es brindar un contexto adicional en torno al uso del sistema y las interacciones entre sus componentes. En el presente documento, cada componente se considera un actor de caso de uso. La sección 4.1 enumera los actores actuales y ofrece una breve descripción de cada uno en el contexto de uso general del sistema. En la sección 4.2, los casos de uso más comunes o generales se describen e ilustran utilizando diagramas de casos de uso de UML y diagramas de secuencia para aclarar las interacciones entre los componentes.

4.1. Actores

Usuario

Este tipo de usuario acorde a su rol, controlará el funcionamiento del sistema, podrá navegar por todas las interfaces gráficas de usuario (GUI) del sistema para realizar operaciones sobre la aplicación

OAuth Authentication Service

El OAuth Authentication Client, es un estándar abierto para la autorización de API's, que nos permite compartir información entre sitios sin tener que compartir la identidad. Es un mecanismo utilizado a día de hoy por grandes compañías como Google, Facebook, Microsoft, Twitter, GitHub o LinkedIn, entre otras muchas.

QueryValidation Service

El objetivo principal de QueryValidation Service es actuar como enlace de comunicación entre un servicio externo (Registro civil) y nuestra aplicación.

Base de datos MySQL

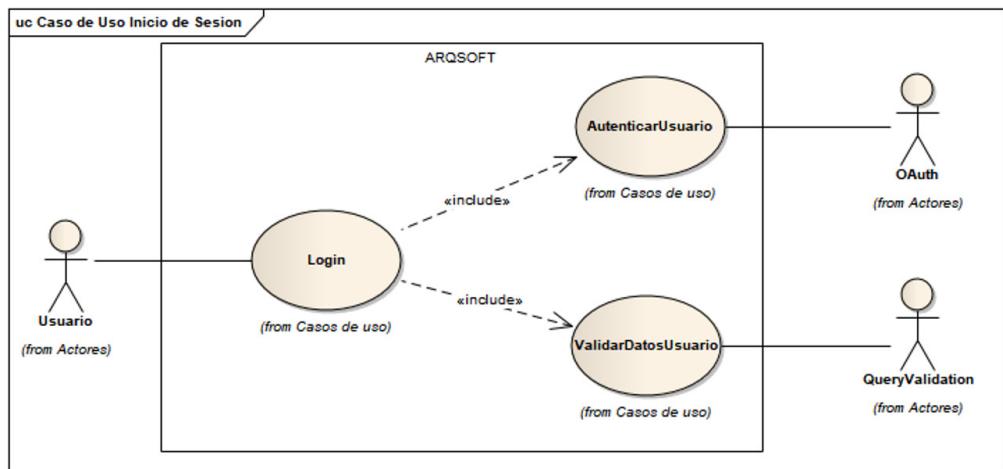
Es el repositorio utilizado para almacenar los datos de la aplicación.

4.2. Realización de casos de uso

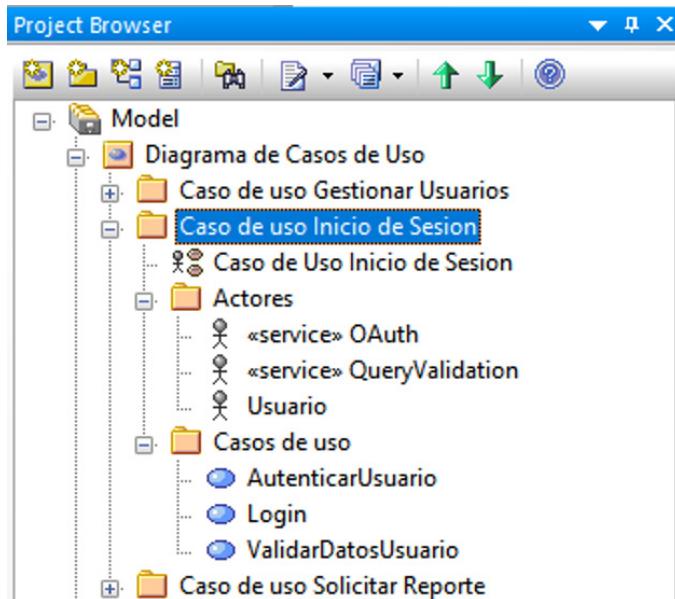
4.2.1. Caso de uso Inicio de Sesión

El usuario debe ingresar las credenciales de acceso (nombre de usuario y contraseña) para realizar el proceso de Login o Logeo. Las credenciales se

autentican y el usuario es redirigido a la página de inicio de la aplicación. Tanto el nombre de usuario y contraseña se validan contra los servicios externos.



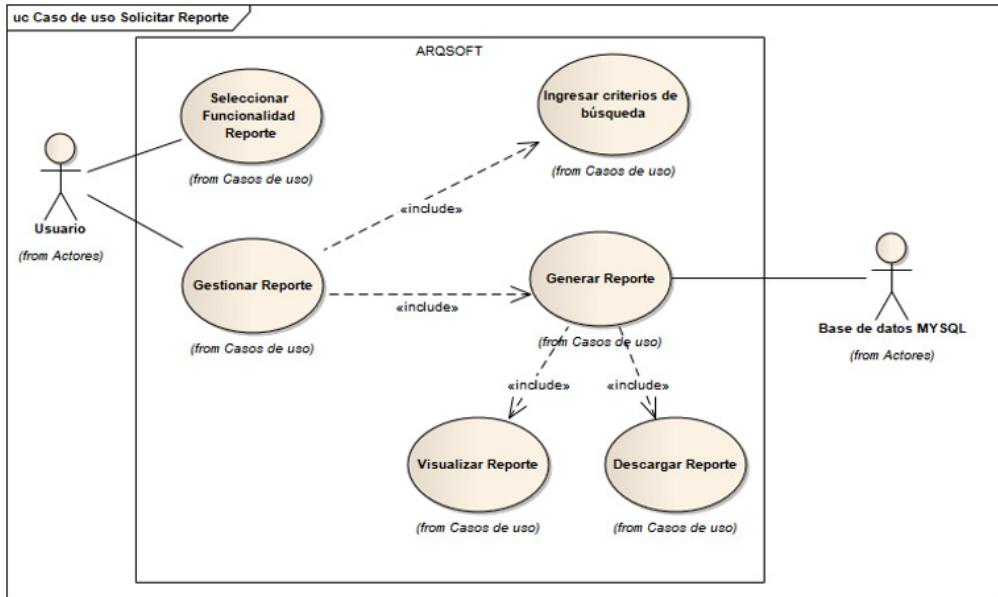
A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



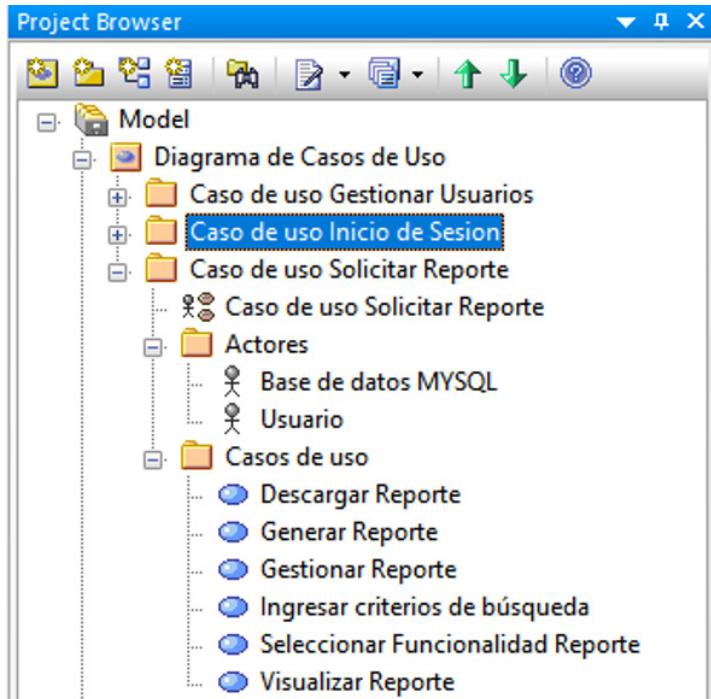
4.2.2. Caso de uso Solicitar Reporte

El usuario solicita a la aplicación un reporte que dependiendo del módulo o funcionalidades a las que tenga acceso podrá recuperar los datos e

información. En este escenario, se ha obviado el proceso que inicia desde el acceso o Login del usuario a la aplicación. Un reporte puede tener varias alternativas de generación, una de ellas por ejemplo que se pueda aplicar criterios de búsqueda o filtros para obtener el reporte. El reporte una vez que es generado por la aplicación puede ser visualizado o descargado por el usuario.

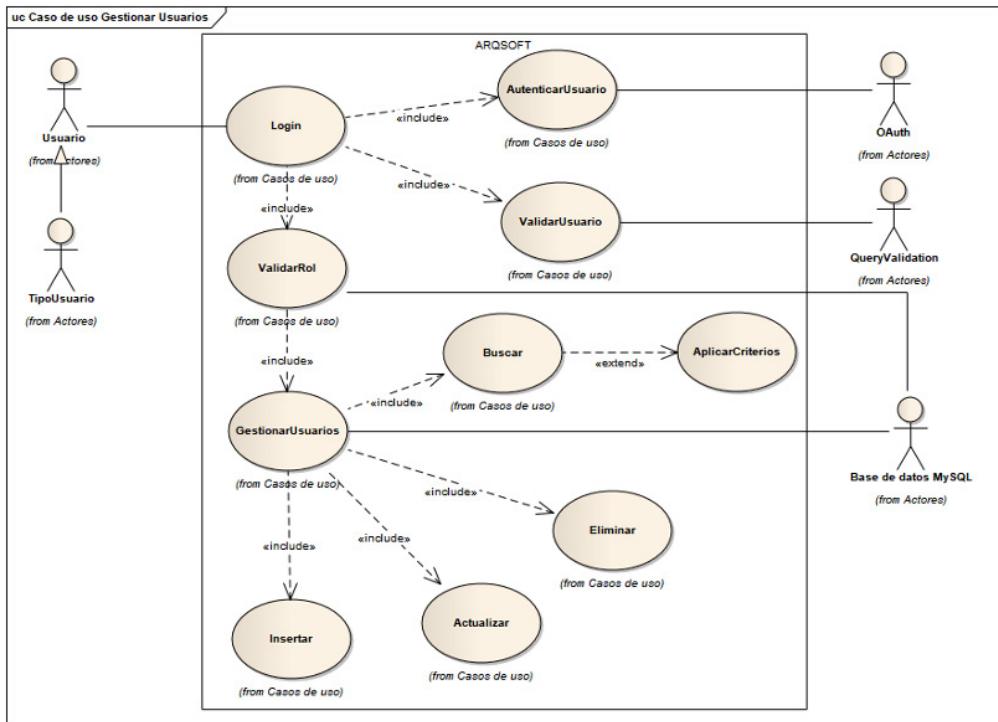


A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)

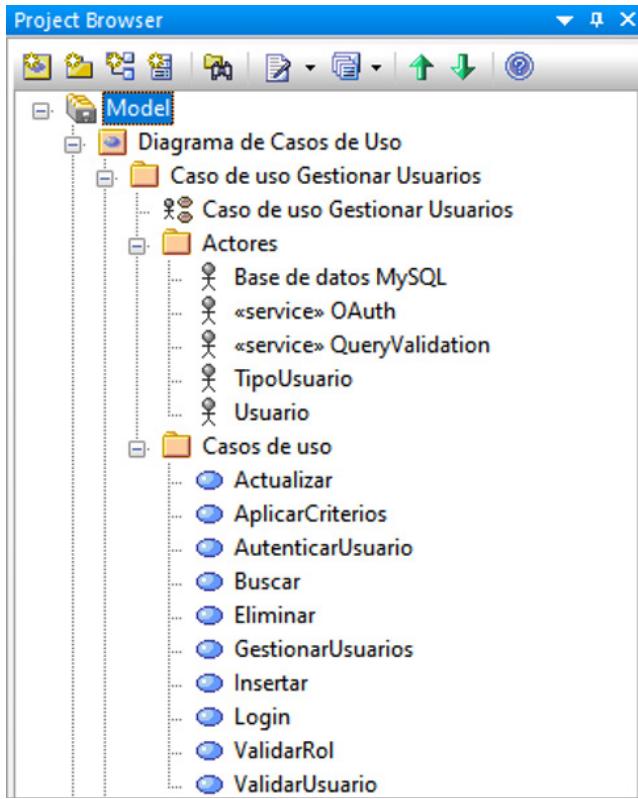


4.2.3. Caso de uso Gestionar Usuarios

El usuario dependiendo del rol que tenga configurado en la aplicación, puede realizar las operaciones CRUD, en este caso de usuarios. para almacenar los datos en la base de datos.



A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



5. Vista de Procesos

5.1. Resumen

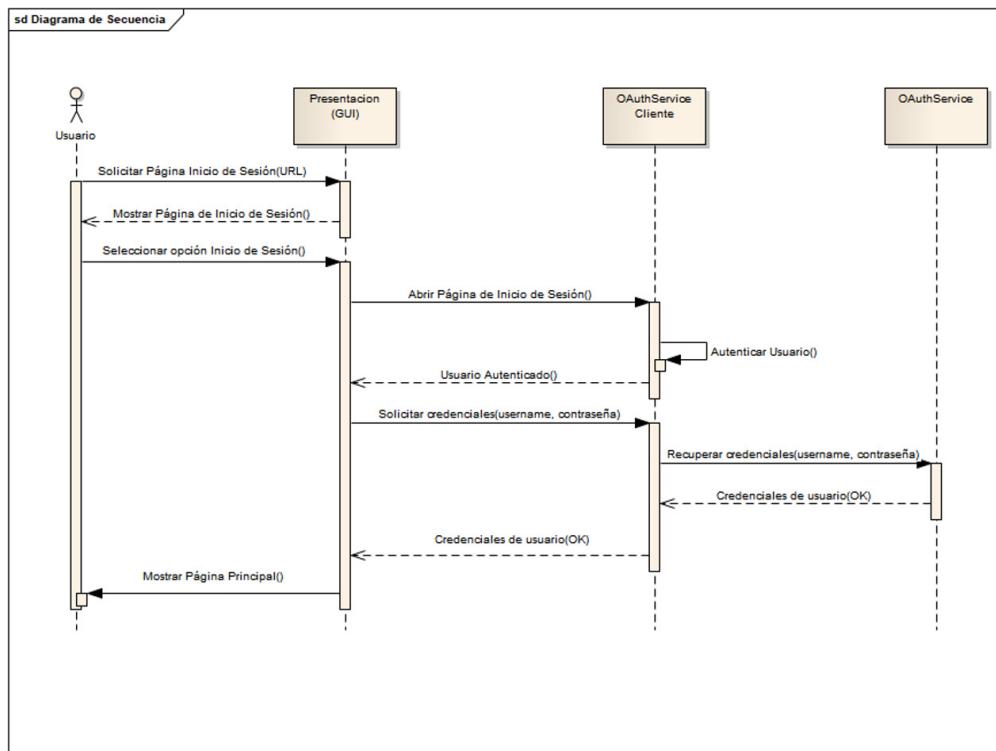
El objetivo de la vista de procesos es capturar tanto el flujo de intercambio de información entre procesos (por ejemplo, llamadas a los API REST o servicios, envío de datos a la base de datos) y la secuencia y el tiempo de estas comunicaciones entre procesos. Esto nos permite considerar cuestiones como la concurrencia y la confiabilidad.

5.2. Realización de diagrama de secuencia

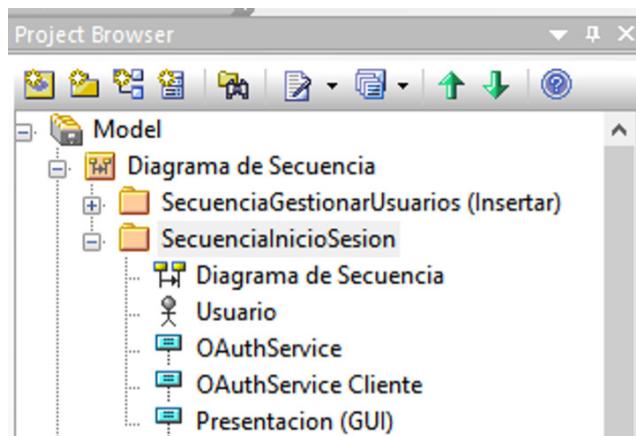
5.2.1. Diagrama de secuencia inicio de sesión usando el servicio OAuth

El usuario primero realiza la petición de la página para realizar la autenticación, para ello ingresa la URL con la dirección donde se encuentra implementada la aplicación. Posterior a ello y cuando se le visualiza la pantalla de inicio de sesión ingresa las credenciales de acceso (nombre de usuario y contraseña) para realizar el proceso de Logeo. Las credenciales viajan desde el OAuthService Cliente hacia el Servicio que se encuentra

alojado en Cloud. La emisión de la respuesta OK corresponde a que el usuario se ha validado correctamente y a partir de ello se visualizará la pantalla principal de la aplicación.



A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)

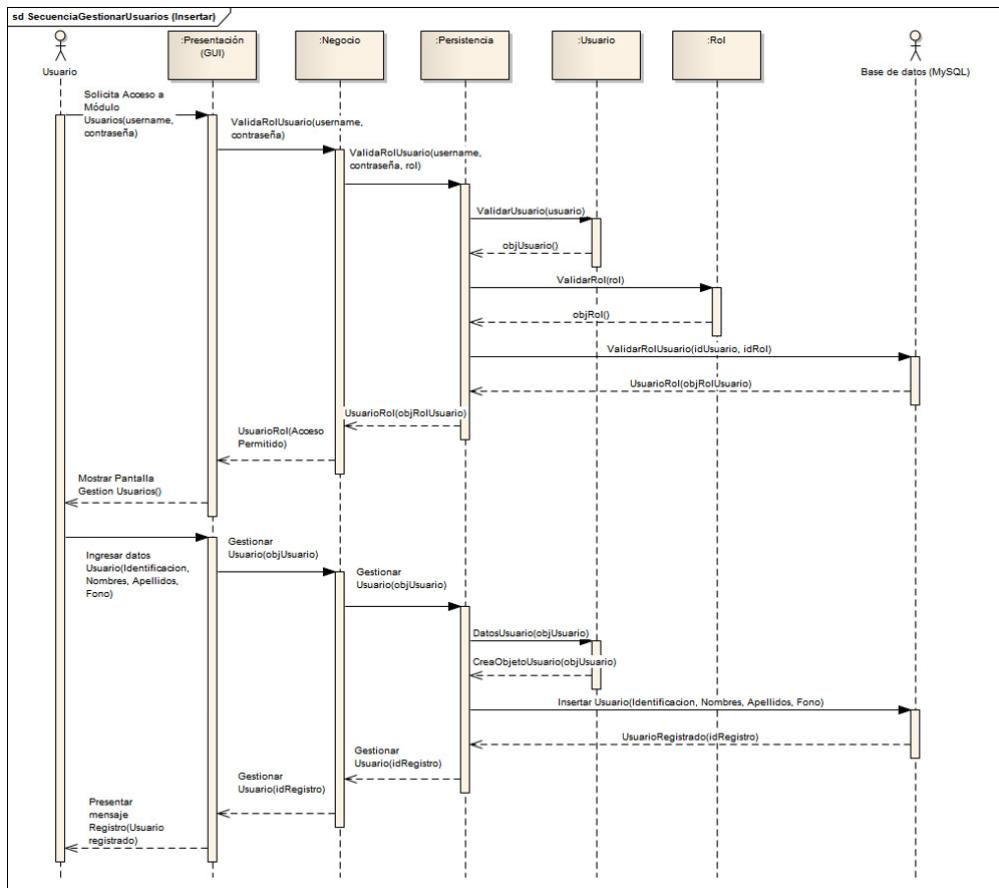


5.2.2. Diagrama de secuencia Gestión de Usuarios (Registrar un Usuario en la Base de Datos)

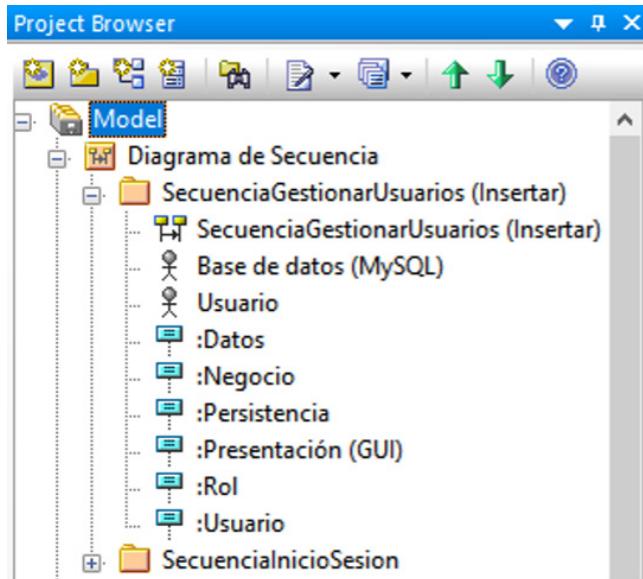
El proceso inicia verificando si el usuario a través de sus credenciales y el rol asociado tiene acceso al módulo y funcionalidades para realizar las operaciones CRUD de Usuarios. Si el usuario tiene acceso al módulo, se muestran los componentes de GUI para ingresar los datos que se soliciten para el registro.

Como se propone el uso de una arquitectura lógica en Capas (3-Layers), los datos se envían desde la capa de Presentación a la lógica de negocio donde se componen o descomponen los objetos para poder operarlos a través de la persistencia.

En la capa de persistencia lo que se realiza es un mapeo de los datos acorde a los objetos, es por ello que intervienen la clase Usuario ya que se crea el objeto, se valida la creación del mismo y posterior a ello los datos son enviados a registrarse en la base de datos. Si el almacenamiento es exitoso, que es el escenario que se muestra en el presente diagrama, los datos se retornan desde la capa inferior hasta la superior que es la de presentación para informar al usuario que los registros se almacenaron correctamente.



A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



6. Vista Lógica

6.1. Resumen

El objetivo principal de la vista lógica es definir los componentes que conformarán el sistema y definir las interfaces a través de las cuales se comunicarán e interactuarán entre sí. El factor de toma de decisiones principal detrás de la definición de los componentes del sistema es la necesidad de aislar los componentes que probablemente cambiarán del resto del sistema. Al definir claramente las interfaces de estos componentes y ocultar sus implementaciones internas del resto del sistema, se puede minimizar el impacto de los cambios esperados. La sección de la Especificación de Requisitos de Software [ERS] describe los cambios que es probable que se realicen en el sistema. Un resumen de estos cambios y cómo la descomposición lógica de la arquitectura los aborda es el siguiente:

1. Cambios en la API de Autenticación

- a. La arquitectura aborda esto implementando las llamadas a la API de autenticación en un componente de OAuthService Client (ver Tabla 3). El resto de la aplicación se comunicará con OAuthService solo a través de la interfaz expuesta por este componente. Por lo tanto, cualquier cambio en el sistema para hacer frente a los

cambios en la API de OAuthService solo debe realizarse en la implementación interna de este componente.

2. Cambios en la API de consulta y validación

- a. De manera similar a lo anterior, esto se soluciona implementando llamadas a la API de consulta de datos y validación de datos e información (*QueryValidation* - <https://www.registrocivil.gob.ec/consulta-de-datos-y-validacion-de-informacion/>) en un componente de cliente del Registro Civil (ver Tabla 3). Los cambios necesarios para hacer frente a los cambios en la API QueryValidation Service solo deben realizarse en la implementación interna de este componente y no en el resto del sistema.

Tabla 3.

Responsabilidades de los elementos

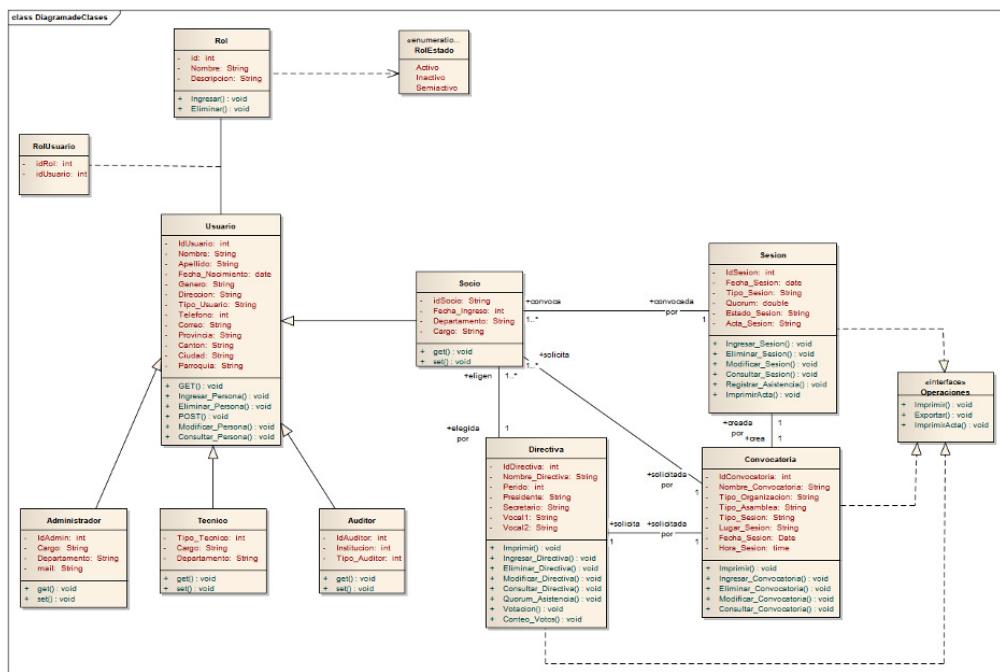
Elemento	Responsabilidades
OAuthService	<ul style="list-style-type: none">▪ Proporcionar una interfaz para la autenticación OAuth.▪ Manejar todas las comunicaciones con la API REST de OAuth▪ Proporcionar una interfaz acorde al lenguaje de programación para que otros componentes la utilicen para acceder a la API de OAuth.
<i>QueryValidation</i>	<ul style="list-style-type: none">▪ Proporcionar una interfaz para la autenticación QueryValidation.▪ Manejar todas las comunicaciones con la API REST de QueryValidation▪ Proporcionar una interfaz acorde al lenguaje de programación para que otros componentes la utilicen para acceder a la API de QueryValidation.
Repositorio de datos (MySQL)	<ul style="list-style-type: none">▪ Almacenar los datos en el almacén de datos MySQL.▪ Proporcionar una interfaz de consulta al repositorio de datos MySQL.
Aplicación ARQSOFT	<ul style="list-style-type: none">▪ Presentar a los usuarios una interfaz de usuario basada en HTML accesible a través de un navegador web.▪ Interactuar con otros componentes del sistema para permitir que los usuarios se autentiquen con Assembla, elijan un proyecto de Assembla para su análisis y analicen el proyecto elegido.

6.2. Realización de diagrama de Clases

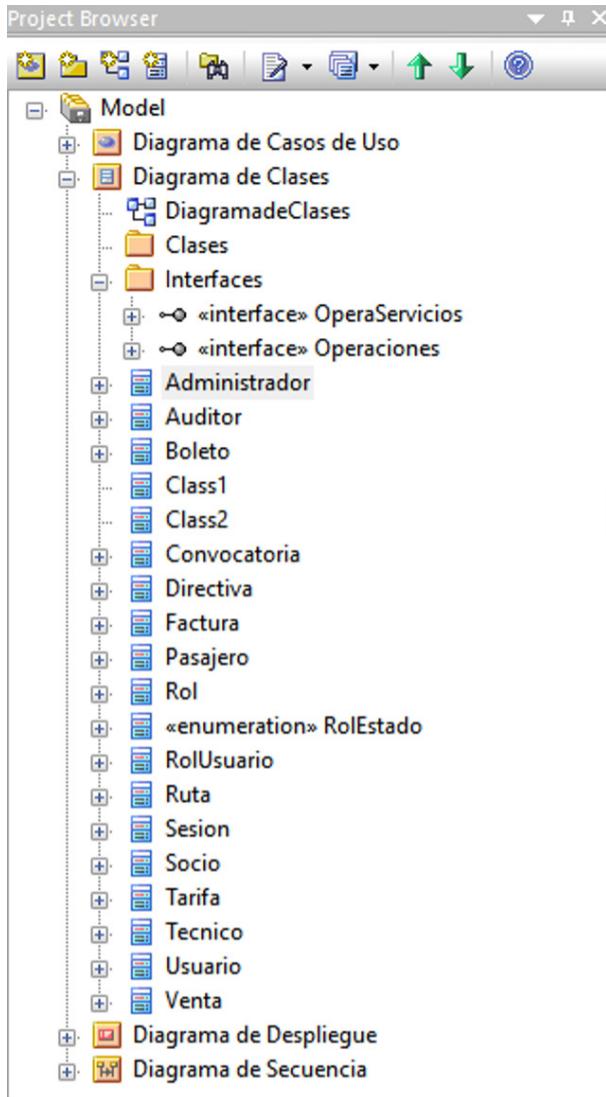
Recuerde que para realizar el diagrama de clases, es fundamental en base al análisis utilizar como herramienta las **Tarjetas CRC para identificación de clases** (<https://creately.com/diagram/example/hmmj51oa2/CRC%20cards>)

Es importante que recuerde que entre las buenas prácticas para elaborar el diagrama de clases siempre nos apoyamos en las tarjetas CRC, ya que por ejemplo si sobre una clase no se puede identificar responsabilidades, es muy probable que esta no sea Clase, además que cuando a partir de la creación de objetos no podamos crear los mismos es porque debemos evaluar si realmente será una clase. Adicional a ello es importante que una clase siempre tenga responsabilidades y colaboradores, de lo contrario será una clase sin ninguna relación y se podría transformar en una Enumeración, por ejemplo.

Nombre de la Clase: Socio	
Responsabilidades	Colaboradores
Crear	Sesión
Actualizar	Convocatoria
Ser parte de una Sesión	Directiva
Ser parte de una Directiva	



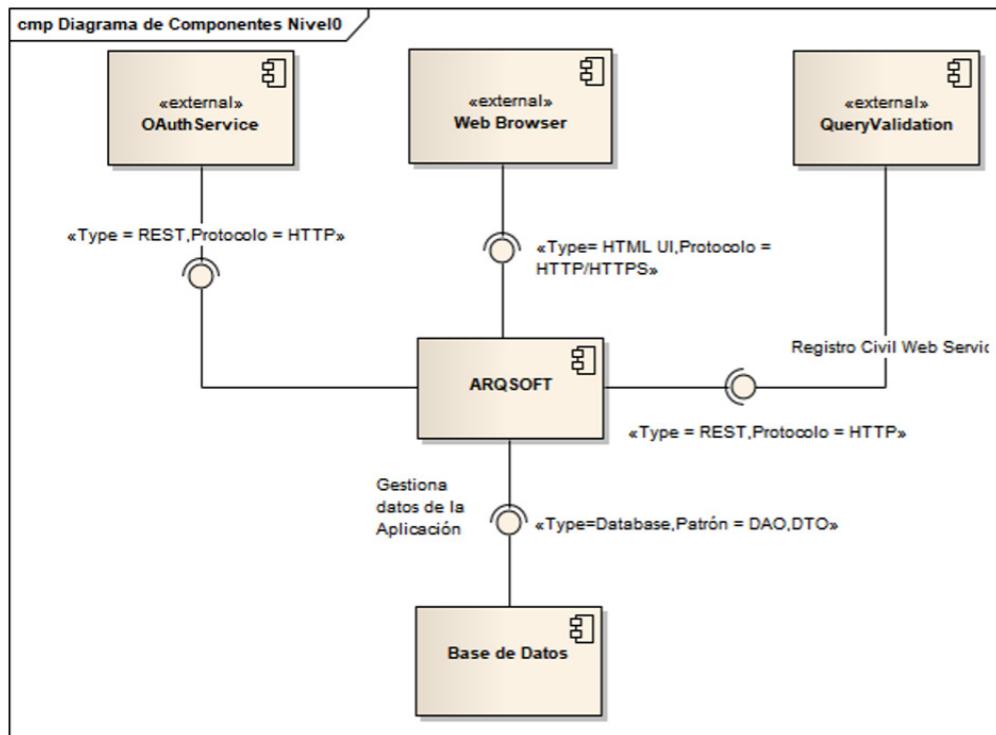
A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



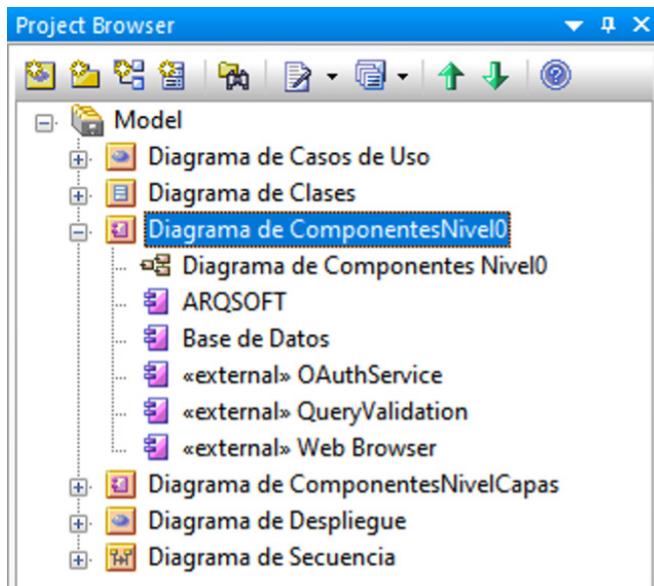
7. Vista de desarrollo

Esta sección describe la estructura general del modelo de implementación, la escritura de la aplicación acorde al patrón arquitectónico y patrones de diseño que se utilice y cualquier componente arquitectónicamente significativo. Por lo general para gestionar el código se hace uso de herramientas tales como Github, Bitbucket.

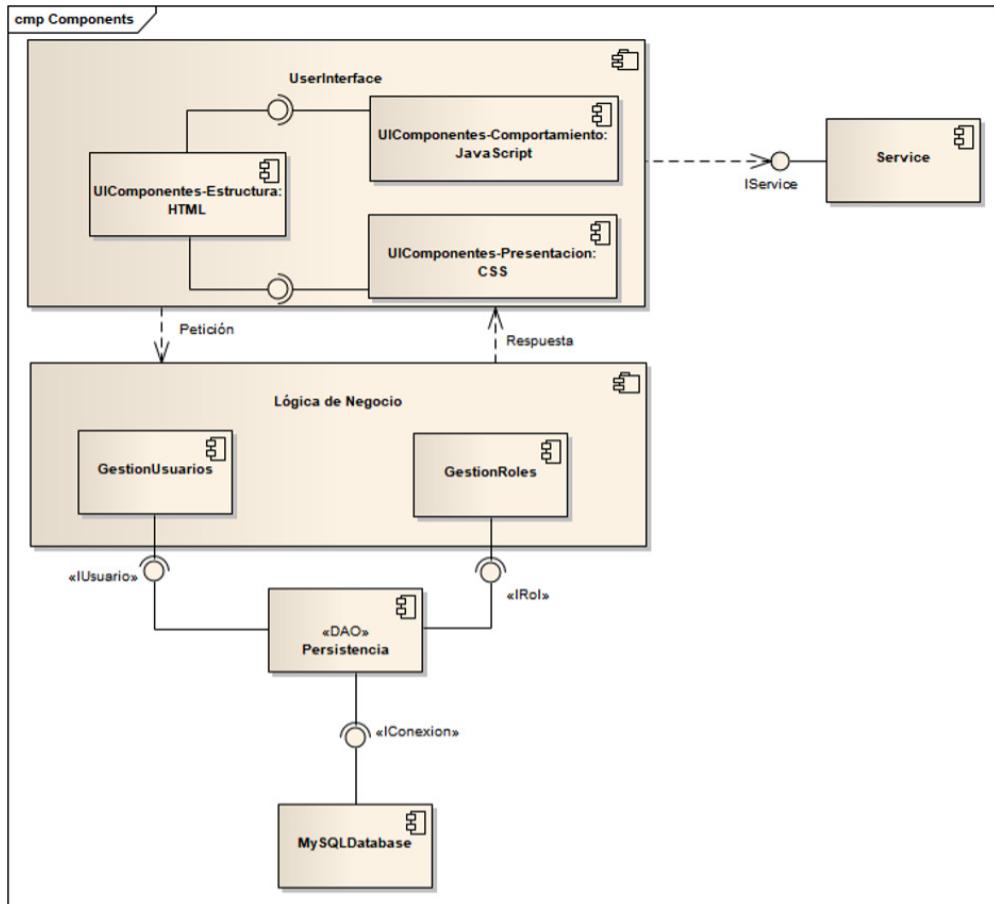
El diagrama de componentes es importante para que los desarrolladores puedan implementarlo a nivel de aplicación. Es por ello por lo que se puede realizar representaciones en alto nivel de abstracción (Diagrama de Componentes Nivel 0) como se muestra en la Figura.



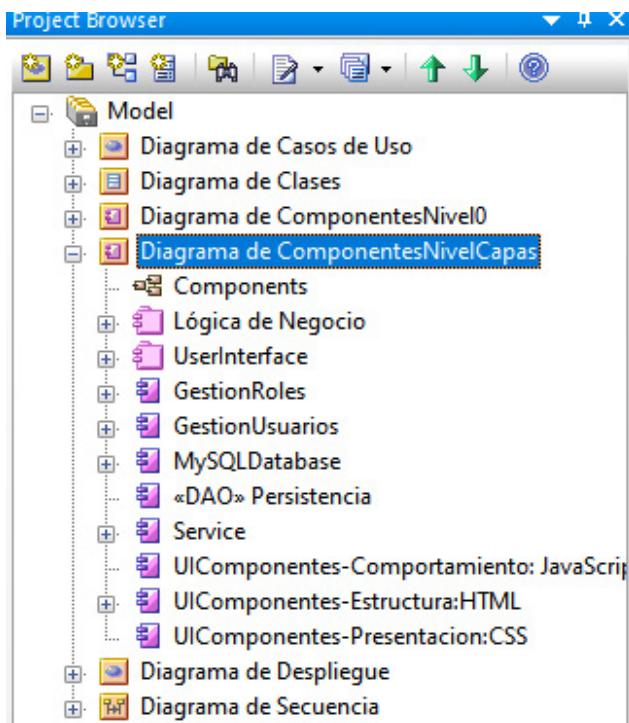
A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



También se puede explorar cada componente de la aplicación para a través de interfaces provistas y requeridas diseñar el comportamiento y la interacción que tendrán los componentes, como se puede ver en la figura donde se expone una representación en Layers (capas lógicas)



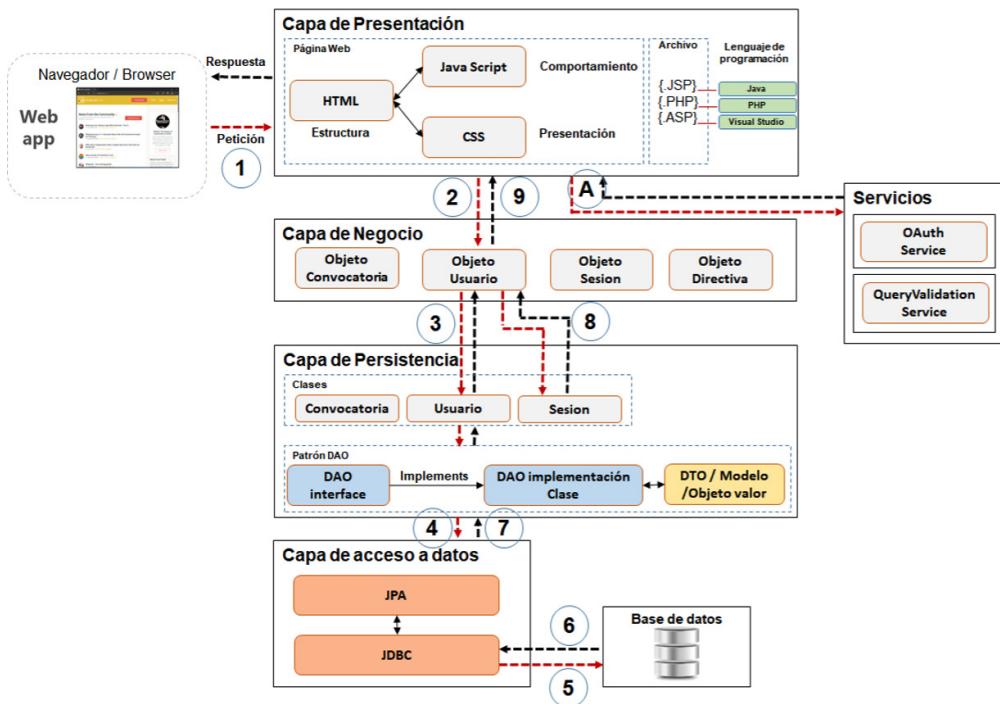
A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



Otra forma de representar la interacción de los componentes que formarán parte de la solución y que ayudará a comprender a los desarrolladores los módulos a implementar a nivel de aplicación, es como se muestra en la figura, donde se indica que la petición dada por un usuario inicia cuando ingresa a través de teclado una petición, dicha petición en una arquitectura de Layers inicia en la capa de presentación en donde a través de la página HTML que está conformada por componentes de UI, y librerías que manejan el comportamiento (Java Script) y estilos como CSS permite enviar la petición a la capa de negocio.

La capa de negocio es la encargada de receptar la petición para agrupar o descomponer el objeto utilizando estructuras de datos, métodos y funciones que permitan aplicar las reglas de negocio y la lógica necesaria para enviar a procesar los datos en la base de datos. Cuando los datos se mapean en la capa de persistencia, en este caso utilizando patrones de diseño como DAO y DTO, el paso de los datos a su procesamiento en la base de datos es muy sencillo, y aunque se puede hacer uso de JPA o Hibernate, dependiendo del lenguaje y tecnología de programación que se utilice, esto ayudará a que los datos mantengan la persistencia lo que quiere decir que si se envía a almacenar en la base de datos un conjunto de datos por ejemplo A, en la base de datos se almacene A y no por ejemplo B,

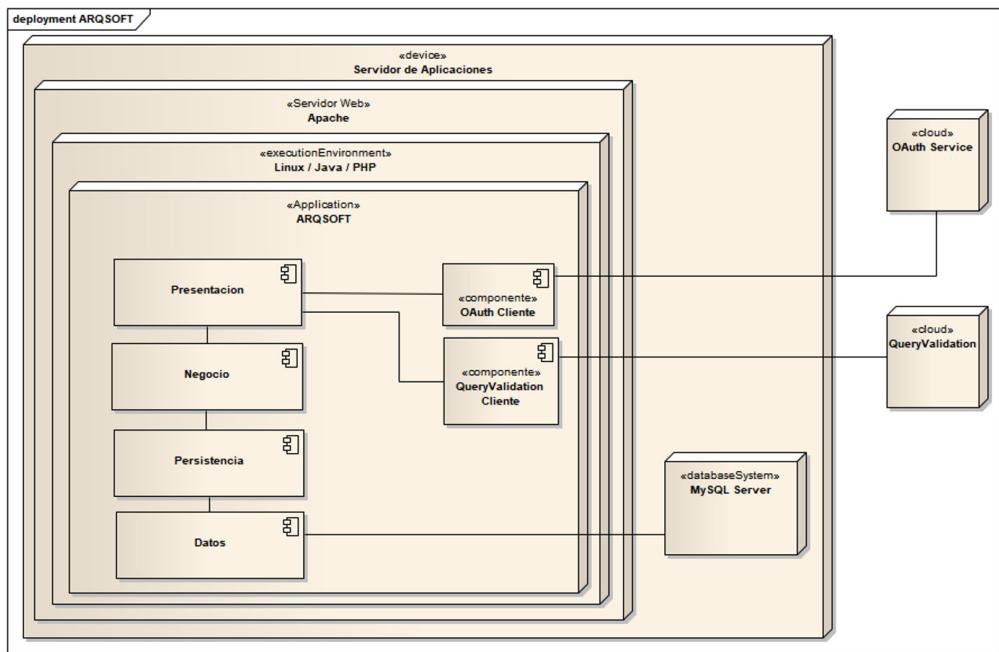
C o A1. En un modelo en capas, el proceso es de dos vías, esto quiere decir que cuando se almacenen los datos en la base de datos deberá existir una respuesta que debe pasar por cada una de las capas como se visualiza en la figura, de lo contrario estaremos violando las reglas de la arquitectura ocasionando inconsistencias.



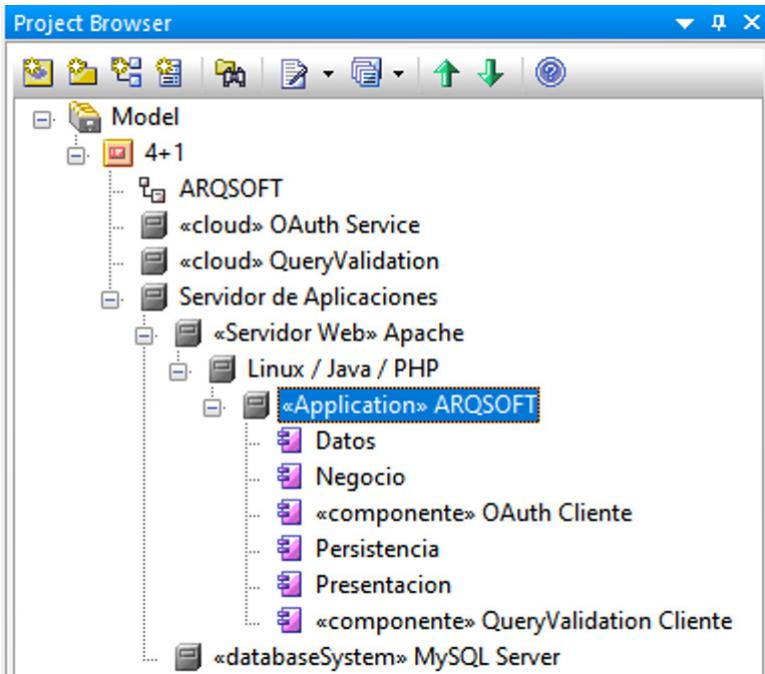
8. Vista de implementación

La aplicación web se alojará en un único servidor físico. Se utilizará un servidor web Apache que permita visualizar las páginas de la aplicación. Además, una instancia de MySQL Server también se alojará en el servidor físico para ayudar a la aplicación a conservar los datos. La aplicación interactuará con API externas (OAuth Service y QueryValidation), de las cuales se desconocen los escenarios de implementación ya que son externos a la aplicación y nuestra aplicación solo debe consumir las API que en base a una petición emiten una respuesta en formato JSON.

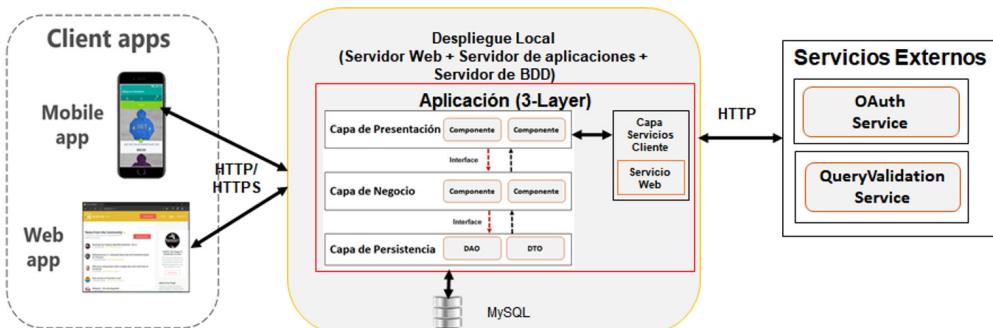
Los detalles de la implementación o despliegue de la aplicación se pueden ver a continuación.



A continuación, se muestra la estructura de representación en la herramienta Enterprise Architect (herramienta CASE disponible en el Laboratorio Virtual UTPL)



Otra forma en la cual podemos exponer la propuesta de implementación o despliegue de la solución es como se muestra en la figura, con lo cual las partes interesadas pueden entender la propuesta arquitectónica.



ANEXO 3: Documento de Especificación de Requisitos de Software

Software Requirements Specification

(Documento de Especificación de Requisitos de Software)

Proyecto ARQSOFT

Elaborado por: Daniel Alejandro Guamán Coronel

Versión 1.0

12/03/2021

Historial de revisiones

Versión	Descripción de versiones / cambios	Responsable	Fecha
1.0	Versión inicial	Daniel Alejandro Guamán Coronel	12/03/2021

Bloque de aprobación

Versión	Comentarios	Responsable	Fecha
1.0	En revisión		12/03/2021

Tabla de contenidos

- 1. Introducción
 - 1.1. Propósito
 - 1.2. Alcance
 - 1.3. Personal involucrado
 - 1.4. Definiciones, acrónimos y abreviaturas
 - 1.5. Referencias
 - 1.6. Resumen
- 2. Descripción General
 - 2.1. Perspectiva del producto
 - 2.2. Características de los Usuarios
 - 2.3. Restricciones
 - 2.4. Suposiciones y dependencias
- 3. Requisitos específicos
 - 3.1. Requerimientos Funcionales
 - 3.2. Requerimientos No Funcionales
- 4. Requisitos comunes de las Interfaces
 - 4.1. Interfaces de Usuario
 - 4.2. Interfaces de Hardware
 - 4.3. Interfaces de Software
 - 4.4. Interfaces de Comunicación
 - 4.5. Requisitos de Rendimiento
 - 4.6. Requisitos de Seguridad
 - 4.7. Requisitos de Usabilidad
 - 4.8. Requisitos de Disponibilidad
 - 4.9. Requisitos de Mantenibilidad
 - 4.10. Requisitos de Portabilidad

Software Requirements Specification

1. Introducción

El presente documento se utiliza para la Especificación de Requisitos Software (ERS) del Sistema de Gestión Organizacional para PYMES - ARQSOFT. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones Funcionales de Requisitos de Software ANSI/IEEE 830, 1998.

1.1. Propósito

El propósito del documento es definir las especificaciones funcionales, no funcionales para el desarrollo del sistema que permita automatizar los procesos de Gestión Organizacional (Gestión de Roles, Permisos y Usuarios (RPU), Gestión de Convocatorias, Gestión de Directiva, Gestionar de Sesiones, Gestión de Socios).

1.2. Alcance

Esta especificación de requisitos está dirigido a los arquitectos, desarrolladores y demás partes interesadas que deseen conocer el Sistema ARQSOFT, el cual tiene por objetivo principal automatizar los procesos de Gestión Organizacional (Gestión de Roles, Permisos y Usuarios (RPU), Gestión de Convocatorias, Gestión de Directiva, Gestionar de Sesiones, Gestión de Socios).

1.3. Personal involucrado

Nombre	Juan Pérez
Rol	Responsable del Instituto de Economía Popular y Solidaria
Categoría Profesional	Gerente General
Responsabilidad	Coordinar las visitas a las Pymes de la Región 7 Loja
Información de contacto	daguaman@utpl.edu.ec

Nombre	Andrea Villegas
Rol	Jefe de Administración y Finanzas
Categoría Profesional	Contadora de la PYME Ecolacteos de Loja
Responsabilidad	Contabilidad General de la Empresa

Nombre	Andrea Villegas
Información de contacto	abc@utpl.edu.ec

Nombre	Ana Carrillo
Rol	Jefe de Operaciones
Categoría Profesional	Recursos Humanos
Responsabilidad	Organizar el personal de logística, asesorías y servicios
Información de contacto	abc@utpl.edu.ec

Nombre	Roberto Méndez
Rol	Presidente de socios de la PYME
Categoría Profesional	Ingeniero Comercial
Responsabilidad	Organizar a los socios de la PYME
Información de contacto	abc@utpl.edu.ec

Nombre	Daniel Alejandro Guamán Coronel
Rol	Analista
Categoría Profesional	Analista de Sistemas
Responsabilidad	Análisis y diseño del Sistema
Información de contacto	daguaman@utpl.edu.ec

1.4. Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
ARQSOFT	Sistema de Gestión Organizacional para PYMES
ERS	Especificación de Requisitos de Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
RPU	Roles, Permisos y Usuarios
CRUD	Create, Read, Update, Delete
Gestión	Término que se usa para especificar que el sistema permitirá realizar operaciones CRUD

1.5. Referencias

Título del Documento	Referencia
Estándar IEEE 830 – 1998	IEEE [IEEE Computer Society. Software Engineering Standards Committee, & IEEE-SA Standards Board. (1998). <i>ieee recommended practice for software requirements specifications</i> (Vol. 830, No. 1998). IEEE.]

1.6. Resumen

Este documento consta de tres secciones. En la primera se muestra la introducción y se proporciona una visión general de las especificaciones de recursos del sistema. En la segunda sección se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo del sistema, sin entrar en excesivos detalles. En la tercera sección se especifican los requisitos que debe satisfacer el sistema.

2. Descripción General

2.1. Perspectiva del producto

El sistema ARQSOFT, acorde a las restricciones arquitectónicas y tecnológicas iniciales, será un producto desarrollado en lenguaje de programación Java o PHP, haciendo uso de Frameworks como Spring Tool Suite o Laravel respectivamente. El sistema debe contener como módulos RPU, Convocatorias, Sesiones Socios. La gestión de todos los módulos implica la realización de operaciones CRUD a través de la interacción entre el cliente y sistema, base de datos o servicios.

2.2. Características de los Usuarios

Tipo de Usuario	ADMINISTRADOR
Formación	Ingeniero en TI o Afines
Actividades	Tendrá permisos para visualizar y operar sobre todos los módulos y funcionalidades del sistema.

Tipo de Usuario	TÉCNICO 1
Formación	Ingeniero en TI o Afines
Actividades	Tendrá permisos para visualizar y operar sobre los módulos y funcionalidades de: Gestión de Socios, Gestión de Convocatorias, Gestión de Sesiones

Tipo de Usuario	TÉCNICO 2
Formación	Ingeniero en TI o Afines
Actividades	Tendrá permisos para visualizar y operar sobre los módulos y funcionalidades de: Gestión de Socios

2.3. Restricciones

1. El sistema está diseñado como una prueba de concepto para un sistema de gestión de datos e información principalmente para las Pymes del Ecuador que debe ser desplegado en la Web, sin embargo, puede escalar para adicionar nuevos módulos, funcionalidades o servicios que se construirán en el futuro. Por lo tanto, uno de los principales interesados en este documento y el sistema en su conjunto son los actuales y futuros arquitectos y diseñadores, no necesariamente los usuarios como suele ser el caso. Como resultado, uno de los objetivos de este documento es que pueda ser útil para los arquitectos, diseñadores y partes interesadas.
2. El sistema debe permitir comunicarse con varias API de sistemas, subsistemas o servicios de terceros. En este caso usaremos OAuth y los servicios expuestos por el Registro Civil para validar datos (<https://www.registrocivil.gob.ec/web-service-2/>). Definir cómo el sistema interactúa con estos sistemas de terceros es una preocupación principal de la arquitectura.
3. El sistema se construirá utilizando lenguajes de programación Java o PHP, y frameworks como Spring Tool Suite o Laravel. utilizará un sistema RDBMS (Sistema de gestión de base de datos relacional) de código abierto (MySQL) para la persistencia de datos y se implementará en un servidor web Linux. Estos requisitos especiales de implementación requieren una consideración adicional en el desarrollo de la arquitectura.
4. El cliente requiere ver entregas continuas y funcionales de los prototipos de la solución para verificar su avance en el diseño y construcción, por tanto considere adoptar una metodología de desarrollo que satisfaga esta necesidad.

2.4. Suposiciones y dependencias

El equipo de arquitectos y desarrolladores de software tienen experiencia en el uso de tecnologías actuales. En los equipos computacionales en el que se implemente el sistema debe cumplir con las características antes mencionadas para el correcto funcionamiento del mismo. Se asume que

los requisitos descritos en el documento se aceptan y validan por las partes interesadas. Además, se considera que existe el presupuesto para la realización del producto de software.

3. Requisitos específicos

3.1. Requerimientos Funcionales

Identificación de requerimiento	RF01
Nombre de requerimiento	Autenticación y autorización de usuarios
Características	Los usuarios deberán autenticarse para acceder a cualquier módulo o funcionalidad del sistema ARQSOFT
Descripción del requerimiento	<p>La aplicación debe verificar a través de las credenciales de acceso (usuario y contraseña) si una persona está registrada en la base de datos de la aplicación y está autorizada para el acceso. Para ello el usuario debe:</p> <ul style="list-style-type: none">Ingresa el nombre de usuarioIngresa la contraseñaLa aplicación deberá validar las credenciales de accesoSi las credenciales de acceso son correctas y el usuario se encuentra registrado en la base de datos, se autoriza su acceso y podrá visualizar la pantalla principal.
Requerimiento No Funcional	
Prioridad del requerimiento:	Alta

Identificación de requerimiento	RF02
Nombre de requerimiento	Autenticación de Usuario contra servicio OAuth
Características	Los usuarios deberán autenticarse a través de un servicio OAuth.
Descripción del requerimiento	<p>La aplicación debe verificar a través de las credenciales de acceso (correo y contraseña) si una persona está autorizada para el acceso.</p> <p>Para ello el usuario debe:</p> <ul style="list-style-type: none">Ingresa la dirección de correo electrónico.Ingresa la contraseña del correo electrónico.Las credenciales de acceso se validarán contra el servicio <i>Authentication Client</i>. Si el acceso es correcto, el servicio emitirá un código HTTP. 200 que significa autenticación correcta, caso contrario el código 400 que significa autenticación incorrecta.
Requerimiento No Funcional	

Identificación de requerimiento	RF02
--	------

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF03
Nombre de requerimiento	Gestión de Usuarios
Características	El usuario con rol de Administrador podrá dar de alta o baja a los usuarios que utilizarán la aplicación.
Descripción del requerimiento	<p>La aplicación debe permitir a través de una GUI, realizar las operaciones CRUD de los usuarios, para ello el usuario con rol Administrador ingresará datos como: Tipo de Identificación, Número de identificación, Nombres, Apellidos, correo electrónico, usuario y contraseña. Algunas consideraciones de los campos usuario y contraseña son:</p> <p>El texto del campo de usuario debe tener una longitud máxima de 8 caracteres y debe permitir letras en mayúsculas y minúsculas.</p> <p>El texto del campo de contraseña debe contener caracteres alfanuméricos y su longitud</p>

Requerimiento No

Funcional

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF04
Nombre de requerimiento	Gestión de Roles
Características	El usuario con rol de Administrador podrá dar de alta o baja a los roles que servirán para asignar los permisos a los usuarios.
Descripción del requerimiento	<p>La aplicación debe permitir registrar los roles que se les podrá asignar a los usuarios a través de la GUI. Para registrar los roles se les solicitará:</p> <p>Nombre del rol (Administrador, Técnico1, Técnico 2, Auditor)</p> <p>Descripción del rol</p> <p>Estado del rol (Activo, Inactivo)</p> <p>Asignar y desasignar roles a los usuarios.</p> <p>Generar reportes de roles, permisos y usuarios</p> <p>Generar logs de actividades de asignación / desasignación.</p>

Requerimiento No

Funcional

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF05
Nombre de requerimiento	Gestión de Roles, Permisos y Usuarios
Características	El usuario con rol de Administrador podrá asignar/desasignar los roles y permisos que se les da a los usuarios.
Descripción del requerimiento	<p>La aplicación debe permitir asignar/desasignar los usuarios a uno o varios roles con los cuales un usuario podrá acceder a la aplicación.</p> <p>Los roles que debe tener registrado el sistema previamente son:</p> <ul style="list-style-type: none"> Rol Administrador: se le otorga permisos para visualizar y operar sobre todos los módulos y funcionalidades de la aplicación. Rol Técnico1: se le otorga permisos para visualizar y operar sobre los módulos y funcionalidades para Gestión de Socios, Gestión de Convocatorias, Gestión de Sesiones Rol Técnico2: se le otorga permisos para visualizar y operar sobre el módulo y funcionalidades para la Gestión de Socios Rol Auditor: se le otorga permisos para visualizar y operar sobre el módulo y funcionalidades para la Gestión de Sesiones <p>La aplicación debe generar logs de actividades de asignación / desasignación de usuarios a roles.</p>

Requerimiento No
Funcional

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF06
Nombre de requerimiento	Consultar datos desde QueryValidationService
Características	El usuario con rol de administrador podrá consultar los datos de los usuarios que utilizarán la aplicación.
Descripción del requerimiento	<p>La aplicación debe consultar datos personales registrados en el Sistema del registro Civil, para ello a través de la GUI se le solicitará que el usuario con rol de Administrador cuando vaya a registrar una persona ingrese como dato el número de identificación de la persona.</p> <p>Si la respuesta es correcta a la petición, el servicio Web retornará en un archivo JSON los datos personales de la persona (Identificación, nombres, apellidos, género, fecha de nacimiento, estado civil, entre otros que constan en el documento de identificación)</p> <p>Si no se obtiene respuesta a la petición desde el servicio Web, este emitirá un mensaje de error 400.</p> <p>El error 400 debe ser interpretado por la aplicación para informar a través de la GUI al usuario.</p>

Requerimiento No
Funcional

Identificación de requerimiento	RF06
---------------------------------	------

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF07
Nombre de requerimiento	Gestión de Convocatorias
Características	El usuario que tiene permisos para acceder al módulo de convocatorias puede registrar las solicitudes de las convocatorias
Descripción del requerimiento	<p>La aplicación debe permitir a través de una GUI, realizar las operaciones CRUD de las convocatorias.</p> <p>Para crear una Convocatoria es necesario tener previamente configurado los Socios y Directiva a través de los módulos y funcionalidades respectivas.</p> <p>Los datos para crear una Convocatoria son:</p> <p>Tipo de Organización, que puede ser:</p> <ul style="list-style-type: none"> Cooperativa Asociación Organismo Comunitario <p>Tipo de Asamblea, que puede ser:</p> <ul style="list-style-type: none"> Asamblea General Consejo de Administración Consejo de Vigilancia <p>Tipo de Sesión, puede ser:</p> <ul style="list-style-type: none"> Ordinaria Extraordinaria Informativa <p>Lugar</p> <p>Fecha de convocatoria</p> <p>Hora de registro</p> <p>Una convocatoria es creada para llevar a cabo una sola sesión.</p>

Requerimiento No

Funcional

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF08
Nombre de requerimiento	Gestión de Directiva
Características	El usuario que tiene permisos para acceder al módulo para registrar la directiva.

Identificación de requerimiento	RF08
Descripción del requerimiento	<p>La aplicación debe permitir a través de una GUI, realizar las operaciones CRUD de la Directiva para llevar a cabo el proceso de la sesión y convocatorias. Para ello se debe realizar los siguientes pasos:</p> <p>La Directiva regirá por un periodo de tiempo de 1 año calendario.</p> <p>Se debe elegir presidente, secretario y vocales que forman parte de la Directiva.</p> <p>La Directiva debe ser elegida entre los Socios que han sido previamente registrados en el módulo de Socios.</p>
Requerimiento No Funcional	
Prioridad del requerimiento: Alta	

Identificación de requerimiento	RF09
Nombre de requerimiento	Gestión de Sesiones
Características	El usuario que tiene permisos para acceder al módulo para registrar las sesiones.

Identificación de requerimiento	RF09
Descripción del requerimiento	<p>La aplicación debe permitir a través de una GUI, realizar las operaciones CRUD de las Sesiones, para llevar a cabo el proceso de la sesión se debe realizar los siguientes pasos:</p> <p>Para llevar a cabo una sesión debe existir una convocatoria asociada. El usuario debe utilizar la funcionalidad del sistema para registrar asistencia lista de los asistentes a la sesión.</p> <p>Para iniciar la sesión, debe validar la existencia de al menos el 50% de los socios (quórum).</p> <p>En caso de que no exista quórum el estado de la sesión se cambia a postergado.</p> <p>Si la sesión se posterga, se debe editar la fecha de convocatoria y sesión. Finalmente se imprimirá un acta con el estado y observaciones de la sesión.</p> <p>Si la sesión se instala, el secretario debe tomar nota de la sesión a través de la funcionalidad de la aplicación. En esta funcionalidad se puede añadir uno a varios ítems que corresponden a los puntos tratados.</p> <p>Al finalizar la sesión se debe imprimir el documento que contenga los datos de la convocatoria y sesión donde consten los ítems, acuerdos o puntos tratados en la sesión</p> <p>Los datos de cada sesión (actual o anterior) pueden ser recuperados para realizar actividades de visualización o edición.</p>

Requerimiento No
Funcional

Prioridad del requerimiento: Alta

Identificación de requerimiento	RF10
Nombre de requerimiento	Gestión de Socios
Características	En este módulo se puede llevar a cabo la gestión de los socios que son un conjunto de personas que forman parte de una PYME.
Descripción del requerimiento	La aplicación debe permitir a través de una GUI, realizar las operaciones CRUD de los Socios, para llevar a cabo el proceso de la sesión se debe realizar solicitar los datos como Identificación, Nombres, Género, Fecha de Nacimiento, Fecha de ingreso, Provincia, Cantón, Ciudad y Parroquia de residencia de cada socio. La aplicación debe permitir generar reportes de socios usando diferentes filtros o criterios de búsqueda.

Requerimiento No
Funcional

Prioridad del requerimiento: Alta

3.2. Requerimientos No Funcionales

Identificación de requerimiento	RNF01
Descripción del requerimiento	La aplicación debe hacer uso de componentes de Interfaz Gráfica de Usuario (GUI) que permitan que el usuario interactúe de forma intuitiva y sencilla.
Atributo de calidad asociado	Usabilidad
Prioridad del requerimiento:	
	Alta
Identificación de requerimiento	RNF02
Descripción del requerimiento	El sistema debe permitir intercambiar datos con otros sistemas, subsistemas, servicios.
Atributo de calidad asociado	Interoperabilidad
Prioridad del requerimiento:	Alta
Identificación de requerimiento	RNF03
Descripción del requerimiento	Los módulos de la aplicación usados en la autenticación y autorización deben permitir intercambiar datos con servicios externos.
Atributo de calidad asociado	Interoperabilidad
Prioridad del requerimiento:	Alta
Identificación de requerimiento	RNF04
Nombre de requerimiento	La aplicación debe garantizar que los datos de las transacciones realizadas se almacenen en un repositorio o en archivos.
Atributo de calidad asociado	Seguridad
Prioridad del requerimiento:	Alta

Identificación de requerimiento	RNF05
Nombre de requerimiento	Por cada acción realizada sobre los módulos o funcionalidades de la aplicación, se debe considerar el almacenamiento de datos en archivos de logs o en tablas de un modelo de datos. Estos datos deberán contener al menos fecha, hora, módulo, funcionalidad, acción y que se pueda llevar procesos de auditoría.
Atributo de calidad asociado	Seguridad
Prioridad del requerimiento:	Alta
Identificación de requerimiento	RNF06
Descripción del requerimiento	Permitir el correcto funcionamiento de la aplicación con 5, 10 o más usuarios. La aplicación debe funcionar correctamente con un número de hasta 10000 usuarios concurrentes
Atributo de calidad asociado	Escalabilidad
Prioridad del requerimiento:	Alta

4. Requisitos comunes de las interfaces

4.1. Interfaces de Usuario

La Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés) dependiendo del lenguaje de programación estará compuesta por páginas HTML (Java Script, CSS, AJAX) que contienen elementos gráficos como botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada a través de un navegador (Browser – Mozilla Firefox, Chrome, Safari u otro).

4.2. Interfaces de Hardware

Las interfaces de Hardware permitirán el desarrollo e implementación local de la aplicación, para ello se propone utilizar:

- 4 equipos computacionales portátiles que contiene software especializado
- 1 servidor con las siguientes características:

Marca: Dell

Modelo: Poweredge 1950 III

Procesador: Intel Xeon cuádruple

Memoria: RAM 4 GB
Disco Duro: 250 GB
Sistema Operativo: Ubuntu 9.04 Server
Servidor Web: Apache 2.0.3
Motor de bases de datos: MySQL 5.1 Community Server
Administrador de MySQL: phpMyAdmin 3.1.3
Lenguaje de scripting: PHP 5.2.9

4.3. Interfaces de Software

Las interfaces de software permitirán la construcción e implementación de la solución, para ello se propone utilizar:

- Sistema Operativo: Linux.
- Spring Tool Suite
- Larabel Framework
- Navegador o Browser: Mozilla Firefox o Chrome.

4.4. Interfaces de comunicación

Las comunicaciones externas entre servidor(es) de datos, aplicación y cliente(s) del sistema deben hacer utilizando protocolos como TCP/IP, HTTPS o HTTSPS.

Algunas consideraciones relacionadas con los atributos de calidad que la aplicación debe contener se exponen a continuación.

4.5. Requisitos de rendimiento

Garantizar que las peticiones y respuestas entre componentes lógicos de la aplicación y su comunicación con la base de datos u otros servicios no afecte el desempeño de la aplicación, de la base de datos, y el tráfico de red.

4.6. Requisitos de Seguridad

Garantizar la seguridad y confidencialidad de los datos que se generan desde la aplicación o que se transmiten entre la aplicación y otros servicios. Bajo este contexto, es necesario evitar la intrusión de personas no autorizadas utilizando certificados digitales de seguridad, firmas digitales, firewalls o algoritmos de encriptación de los datos considerables sensibles para la organización.

4.7. Requisitos de Usabilidad

La aplicación web y cuando se considere adicionar una aplicación móvil, deben ser diseñadas y construidas a nivel de Front - End utilizando componentes de GUI que permitan que la aplicación sea intuitiva y sencilla, visualmente agradable a la vista y tenga en cuenta los principios de UX (Experiencia de Usuario) y UI (Interfaz de Usuario).

4.8. Requisitos de Disponibilidad

La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios las 24 horas los 7 días de la semana, los 365 días del año, garantizando un esquema adecuado que permita ante un posible fallo en cualquiera de los componentes de la aplicación, contar con una contingencia, generación de alarmas.

4.9. Requisitos de Mantenibilidad

El sistema debe disponer de documentación actualizada a nivel planificación, análisis, diseño, construcción e implementación que permita realizar operaciones de mantenimiento con el menor esfuerzo posible para las partes interesadas.

4.10. Requisitos de Portabilidad

El sistema debe estar diseñado para ejecutarse o desplegarse en hardware que tenga unas características a nivel de software (Sistema Operativo, Navegadores Web) para el correcto funcionamiento de la aplicación del sistema operativo que se ejecute.