

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	<i>Frameworks para desarrollo en web (Tronco común)</i>
Carrera:	Ingeniería en Tecnologías de Información y Comunicaciones e Ingeniería en Sistemas Computacionales
Clave de la asignatura:	CMC-1301
Créditos ¹	2-2-4

2.- PRESENTACIÓN

Caracterización de la asignatura.

Esta asignatura tiene como propósito capacitar a los alumnos de Ingeniería en Sistemas Computacionales e Ingeniería en Tecnologías de Información en el uso de *frameworks* que disminuyen el tiempo invertido en la construcción y modificación de *software* en el ambiente *web*, para responder oportunamente y pertinentemente a los requerimientos de la comunidad de disponer de sistemas de información apropiados en tiempos relativamente cortos.

Dada la variedad que existe de estas herramientas, se centra en desarrollar habilidades para *SPRING* y *DJANGO*, los cuales pueden ser operados desde *PYTHON* y *JAVA*, que son dos de los lenguajes de programación empleados en la formación de los alumnos de ambas carreras en esta institución.

Consta de tres unidades. La primera brinda una introducción al ambiente y características de estas herramientas. La segunda y la tercera se centran en el desarrollo de habilidades en el alumno para la construcción de software mediante *DJANGO* y *SPRING*, respectivamente. **Estas habilidades son aplicadas para construir una aplicación básica que pueda ser empleada para resolver necesidades de alguna organización o para permitirle al estudiante emplear su creatividad en la oferta de servicios de Internet a los internautas.**

En estudiantes de Ingeniería en Sistemas Computacionales contribuye a las siguientes competencias especificadas en el perfil de la retícula genérica:

- Desarrollar, implementar y administrar software de sistemas o de aplicación que cumpla con los estándares de calidad con el fin de apoyar la productividad y competitividad de las organizaciones.
- Diseñar, desarrollar y administrar bases de datos conforme a requerimientos definidos, normas organizacionales de manejo y seguridad de la información, utilizando tecnologías emergentes.
- Integrar soluciones computacionales con diferentes tecnologías, plataformas o

¹ Sistema de asignación y transferencia de créditos académicos

dispositivos.

Asimismo, en los alumnos de Ingeniería en Tecnologías de Información y Comunicaciones impulsa las competencias profesionales que se señalan a continuación:

- Desarrollar e implementar sistemas de información para el control y la toma de decisiones utilizando metodologías basadas en estándares internacionales.
- Administrar proyectos que involucren Tecnologías de la Información y Comunicaciones en las organizaciones conforme a requerimientos establecidos.
- Desempeñar funciones de consultoría y auditoría en el campo de las Tecnologías de la Información y Comunicaciones.
- Utilizar tecnologías y herramientas actuales y emergentes acordes a las necesidades del entorno.
- Diseñar, desarrollar y mantener sistemas de bases de datos asegurando la integridad, disponibilidad y confidencialidad de la información almacenada.

En ambas carreras es la asignatura que continúa con el esfuerzo iniciado por “Programación Web” para que el alumno desarrolle sus habilidades de programación en el ambiente *web*.

Intención didáctica.

Esta asignatura consta de cuatro unidades orientadas al desarrollo de habilidades para emplear los marcos trabajo (*frameworks*) en el desarrollo de *software*, buscando proporcionar a los futuros egresados herramientas que les permitan construir aplicaciones en el ambiente *web* en forma oportuna y pertinente, para responder a la exigencia de los clientes de poseer sistemas de información que involucren el menor tiempo posible en su construcción. Su intención no sólo queda allí. También, busca que los futuros profesionistas hagan rápidamente cambios a las aplicaciones, debido a que los clientes pueden aumentar sus requisitos o modificarlos cuando las ven funcionando.

Está proyectada para que el alumno, en equipo de trabajo, construya una aplicación básica que pueda emplearla, enriqueciéndola posteriormente, en la residencia profesional, el servicio social o para promover los productos o servicios de alguna empresa a la que haya formulado su plan de negocio en las actividades de emprendurismo que él haya decidido realizar. Por tal motivo, se sugiere al catedrático que, al inicio del curso, les pida efectuar el planteamiento de construcción de una aplicación con estas características para las unidades dos y tres.

La unidad uno tiene como propósito brindar los conceptos, razones y evolución de los *frameworks*, con la finalidad de que los alumnos comprendan y apliquen apropiadamente la filosofía, los conceptos y las arquitecturas en que se sustentan los dos marcos de trabajo proporcionados en la unidades dos y tres.

Las unidades dos y tres se centran en el estudio y aplicación de *DJANGO* y *SPRING*, respectivamente. Ambos fueron seleccionados por que pueden vincularse con los lenguajes de programación, *PYTHON* y *JAVA*, que se emplean en los cursos de programación de las carreras Ingeniería en Sistemas Computacionales e Ingeniería en Tecnologías de

Información y Comunicaciones ofertadas por el Instituto Tecnológico de Chetumal. Ambas unidades se han diseñado aplicando la política “Comprender los conceptos y después aplicarlos”. Por ese motivo, en las unidades de aprendizaje se pide la elaboración de resúmenes, cuadros sinópticos, mapas mentales o mapas conceptuales, para que, posteriormente, se le solicite al alumno que construya una aplicación para ambiente *web*, haciendo uso del *framework* estudiado en cada una de ellas. La forma de abordar la parte práctica de ambas unidades es variada, por lo tanto, el catedrático podría optar por alguna de las siguientes estrategias de aprendizaje:

1. Emplear un caso de estudio que irá resolviendo conforme avanza en la unidad para que, al final de ella, solicite a los alumnos resolver sus propios casos de estudio haciendo uso de los conocimientos y habilidades desarrollados.
2. Emplear un caso de estudio que irá resolviendo parcialmente conforme avanza la unidad y solicitarle a los alumnos que le incluyan características no consideradas por él al momento de la impartición del curso.
3. Realizar ejercicios donde se muestre cómo hacer uso de las características y capacidades del *framework* estudiado, pidiéndole a los estudiantes que apliquen lo aprendido en los casos de estudio que previamente les ha asignado.
4. Emplear un caso de estudio en el estudio del *framework* y solicitarle a los alumnos que usen lo aprendido en el desarrollo de la aplicación que tendrán que entregar al final de la unidad.

Éstas son sólo cuatro estrategias, sin embargo, el catedrático podría encontrar varias más, motivo por el cual no se le coarta en su creatividad, pero, si se le pide que dosifique las actividades de los alumnos para no abrumarlos con las actividades realizadas en esta asignatura y las efectuadas en otras. De hecho, se le solicita que se ponga de acuerdo con profesores de asignaturas que estén cursando los alumnos para idear la realización de proyectos integradores que pudieran desarrollar los estudiantes, con el objetivo de no saturarlos de actividades desarticuladas con objetivos carentes de visión sistémica. Otro requisito que se presenta a la creatividad del catedrático es que el alumno debe entregar una aplicación que resuelva un problema real, desarrollada en el *framework* estudiado.

Entre las prácticas propuestas, la número uno es desglosada con la finalidad de brindar una idea al catedrático de la forma en que debe especificar los casos de estudio que solicite a los alumnos que realicen.

Las actividades de aprendizaje y las prácticas sugeridas son una guía del conjunto de actividades grupales, de equipo o individual que el catedrático puede emplear para alcanzar los objetivos del curso. Esto no restringe su derecho a decidir si incluye actividades nuevas por considerarlas más efectivas o eliminar algunas de ellas si piensa que se está afectando el principio de dosificación. En última instancia, el catedrático deberá tomar la decisión sobre qué incluir o excluir del curso (exceptuando la última actividad de aprendizaje solicitada para las unidades 2 y 3), ya que el conocerá las circunstancias en las cuales se encontrará inmerso cuando se imparta.

3.- COMPETENCIAS A DESARROLLAR

Competencias específicas:	Competencias genéricas
Construir aplicaciones <i>web</i> , empleando marcos de trabajo (<i>frameworks</i>), que resuelvan problemas básicos relacionados con actividades empresariales o con actividades actuales del ámbito de <i>Internet</i> .	1- Competencias instrumentales: <ul style="list-style-type: none">• Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético.• Capacidad de organizar y planificar.• Resolución de problemas.• Toma de decisiones.• Destrezas de computación.• Búsqueda y manejo de información. 2- Competencias interpersonales: <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Habilidad para trabajar en un ambiente laboral. 3- Competencias sistémicas: <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos en la práctica.• Habilidades de investigación.• Capacidad de aprender.• Habilidad para trabajar en forma autónoma.

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Chetumal. Octubre de 2012.	<ul style="list-style-type: none">• Lic. Lino Rangel Gómez• Ing. Carlos Flores Pérez	Diseño de los módulos de especialidad "TIC verde" y "Cómputo móvil y <i>web</i> ".

5.- OBJETIVO GENERAL DEL CURSO

Construir aplicaciones *web*, empleando marcos de trabajo (*frameworks*), que resuelvan problemas básicos relacionados con actividades organizacionales o con actividades actuales del ámbito de *Internet*.

6.- COMPETENCIAS PREVIAS

1. Resolver problemas mediante la elaboración de programas en *PYTHON*, bajo el paradigma orientado a objetos.
2. Resolver problemas mediante la elaboración de programas en *JAVA*, bajo el paradigma orientado a objetos.
3. Desarrollar aplicaciones *web* que involucre lenguajes de marcas, de presentación, del lado del cliente, del lado del servidor, con la integración de servicios *web*.

7.- TEMARIO

Unidad	Temas	Subtemas
1	Introducción	<ol style="list-style-type: none">1. Estrategias de reutilización<ol style="list-style-type: none">1.1. Introducción1.2. Patrones de diseño1.3. Componentes1.4. Marcos de trabajo2. Patrones de diseño<ol style="list-style-type: none">2.1. Elementos de un patrón de diseño2.2. Bibliotecas de patrones3. Componentes<ol style="list-style-type: none">3.1. Elementos de un componente3.2. Bibliotecas de componentes4. Marcos de trabajo<ol style="list-style-type: none">4.1. Elementos del diseño de un marco de trabajo4.2. Cualidades de un marco de trabajo4.3. Biblioteca para un marco de trabajo5. Marcos de trabajo para <i>web</i><ol style="list-style-type: none">5.1. Definición de marco trabajo para <i>web</i>5.2. Evolución5.3. Modelo vista-controlador en los marcos de trabajo para <i>web</i>5.4. Capacidades comunes en los marcos de trabajo de aplicación para <i>web</i>
2	<i>DJANGO</i>	<ol style="list-style-type: none">1. Introducción2. Instalación y configuración

		3. Fundamentos de las páginas <i>web</i> dinámicas 4. Sistema de plantillas 5. Conexión con bases de datos 6. El sitio de administración 7. Procesamiento de formas 8. Configuraciones de <i>URLs</i> y vistas, avanzadas 9. Vista genéricas 10. Seguridad
3	<i>SPRING</i>	1. Introducción 2. Instalación y configuración 3. Conexión a <i>Beans</i> 4. Minimización de la configuración <i>XML</i> 5. Orientación a aspectos 6. Conexión a bases de datos 7. Administración de transacciones a bases de datos 8. Módulo MVC 9. Módulo <i>Web Flow</i> 10. Módulo de seguridad 11. <i>Spring beans</i> con <i>JMX</i>

8.- SUGERENCIAS DIDÁCTICAS

Se recomienda que el profesor:

- Sea conocedor de la disciplina que está bajo su responsabilidad.
- Oriente el trabajo del estudiante, potenciando en él la autonomía, el trabajo cooperativo y la toma de decisiones.
- Muestre flexibilidad en el seguimiento del proceso formativo y propicie la interacción entre estudiantes.
- Impulse a los alumnos a participar con sus proyectos terminados en los concursos de innovación tecnológica que anualmente realiza la Dirección General de Educación Superior Tecnológica.
- Promueva, en los alumnos que hayan realizado proyectos prometedores, a participar en programas de estímulo o apoyo a emprendedores que establecen los gobiernos federal y estatal así como la iniciativa privada.

- Apoye actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Propicie el uso de las nuevas tecnologías en el desarrollo de los contenidos de la asignatura.
- Aliente actividades de planeación y organización de distinta índole en el desarrollo de la asignatura.
- Fomente actividades grupales que favorezcan la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
- Favorezca, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Lleve a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como: observación, identificación manejo y control de de variables y datos relevantes, planteamiento de hipótesis, de trabajo en equipo.
- Desarrolle actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Propicie el uso adecuado de conceptos y de terminología científico-tecnológica
- Proponga problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Relacione los contenidos de la asignatura con el cuidado del medio ambiente; así como con las prácticas de una ingeniería con enfoque sustentable.
- Observe y analice fenómenos y problemáticas propias del campo ocupacional.
- Relacione los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación de la asignatura debe de ser continua y se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en obtener evidencias de aprendizaje tales como:

- Resúmenes, cuadros sinópticos, mapas mentales, mapas conceptuales, planes de mercadotecnia, diseño de producto o servicio, etcétera.
- Solución de casos de estudio o problemas reales que requieren la aplicación de los marcos de trabajo vistos en el curso.
- Participación y desempeño en el aula e investigaciones de campo.

- Dar seguimiento a la comprensión de los conceptos vistos en el desarrollo del temario (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales y de ingeniería).
- Se recomienda utilizar varias técnicas de evaluación con un criterio específico para cada una de ellas (teórico-práctico).
- Uso de una plataforma educativa en *Internet*, la cual puede utilizarse como apoyo para crear el portafolio de evidencias del alumno (integrando: tareas, prácticas, evaluaciones, etc.)

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Introducción

Competencia específica a desarrollar	Actividades de Aprendizaje
Definir el concepto de <i>framework</i> y resumir asuntos tales como su evolución, características comunes, ventajas y desventajas y las razones de su empleo para distinguir los fundamentos, estructura y estrategias de los marcos de trabajo estudiados.	<ol style="list-style-type: none"> 1. Elaborar un mapa mental que defina y relacione los siguientes conceptos: <ol style="list-style-type: none"> 1.1. Estrategia de reutilización. 1.2. <i>Framework</i> 1.3. Componente 1.4. Patrones de diseño 1.5. Elementos del patrón de diseño 1.6. Elementos del <i>framework</i> 1.7. Elementos del componente 2. Crear un cuadro sinóptico que defina y relacione los siguientes conceptos. <ol style="list-style-type: none"> 2.1. Mapeador relacional de objetos (ORM). 2.2. Patrón de diseño modelo-vista-controlador (MVC). 2.3. Patrón de diseño modelo-plantilla-vista (MTV). 2.4. Plantillas HTML. 2.5. <i>XML</i>. 2.6. <i>JSON</i>. 2.7. Prácticas de desarrollo ágil. 2.8. Script <i>CGI</i>. 2.9. <i>PHP</i>.

Unidad 2: *DJANGO*

Competencia específica a desarrollar	Actividades de Aprendizaje
Construir aplicaciones <i>web</i> elementales, haciendo uso de	<ol style="list-style-type: none"> 1. Realizar resúmenes, lógicamente organizados,

<p>DJANGO, siguiendo los principios y paradigmas en que se basa, para resolver problemas típicos del ámbito empresarial o del propio ambiente de <i>Internet</i>.</p>	<p>que conjuntamente indiquen:</p> <ol style="list-style-type: none"> 1.1. Cómo <i>Django</i> emplea modelos para definir datos 1.2. Cómo <i>Django</i> implementa las plantillas 1.3. Cómo <i>Django</i> configura los proyectos 1.4. Cuáles vistas están disponibles en la interface <i>Admin</i>. 1.5. Los pasos a seguir en <i>Django</i> para que pueda acceder a una base de datos. 1.6. Los pasos a seguir para activar un modelo en <i>Django</i>. 1.7. Cómo configurar <i>Django</i> para que acepte peticiones específicas de <i>URLs</i>. 1.8. Cómo crear una vista simple para un navegador <i>web</i>. 1.9. Cómo activar un modelo en la interface <i>Admin</i>. 1.10. Cómo crear relaciones entre clases y modelos de datos. 1.11. Cómo manejar las peticiones <i>HTTP</i>. 1.12. Cómo configurar los patrones <i>URL</i> para habilitar una vista. 1.13. Cómo agregar, acceder objetos en una base de datos <i>SQL</i>. 1.14. Cómo mostrar datos provenientes de la base de datos en una vista. 1.15. Cómo implementar prefijos de vista. 1.16. Cómo configurar el directorio de plantillas para un proyecto. 1.17. Cómo presentar vistas usando plantillas. 1.18. Cómo extender las plantillas. 1.19. Cómo insertar plantillas en otras plantillas. 1.20. Cómo aplicar filtros en las plantillas. 1.21. Cómo actualizar datos de una base de datos usando formas. 1.22. Cómo agregar objetos a la base de datos usando formas. 1.23. Cómo usar vistas genéricas simples.
---	---

	<p>1.24. Cómo emplear las vistas genéricas para crear, actualizar y borrar objetos.</p> <p>1.25. Cómo cargar plantillas desde las aplicaciones.</p> <p>1.26. Cómo crear etiquetas y filtros personalizados.</p> <p>1.27. Cómo crear bibliotecas personalizadas para adecuar el sistema de plantillas.</p> <p>1.28. Cómo crear permisos personalizados en el modelo.</p> <p>1.29. Cómo asignar permisos a usuarios y grupos.</p> <p>1.30. Cómo crear grupos en el sitio <i>web</i> y cómo asignarles usuarios.</p> <p>1.31. Cómo agregar las funcionalidades “<i>login</i>”, “<i>logout</i>”.</p> <p>1.32. Cómo verificar la autenticación y los permisos.</p> <p>1.33. Cómo limitar el acceso a vistas genéricas.</p> <p>1.34. Cómo configurar el administrador de sesiones.</p> <p>1.35. Cómo usar <i>cookies</i> de <i>Django</i> para verificar que el navegador las soporta.</p> <p>1.36. Cómo agregar y recuperar datos de sesiones.</p> <p>1.37. Cómo establecer y recuperar <i>cookies</i>.</p> <p>1.38. Cómo personalizar campos desplegados en la lista de cambios del modelo.</p> <p>1.39. Cómo agregar funcionalidad a la lista de cambios del modelo.</p> <p>1.40. Cómo implementar una jerarquía de datos para navegar en listas de cambios del modelo.</p> <p>1.41. Cómo anular plantillas <i>admin</i> desde los niveles sitio, objeto y <i>block</i>.</p> <p>1.42. Cómo crear vista <i>admin</i> personalizadas.</p> <p>1.43. Cómo anular vistas <i>admin</i> integradas.</p> <p>1.44. Cómo instalar y usar las aplicaciones <i>middleware</i>.</p> <p>1.45. Cómo implementar vistas y peticiones</p>
--	---

	<p>pre-procesadores.</p> <p>1.46. Cómo implementar respuestas y excepciones post-procesadores.</p> <p>1.47. Cómo crear y habilitar un mapa del sitio.</p> <p>1.48. Cómo crear un mapa del sitio para vistas genéricas.</p> <p>1.49. Cómo crear un índice para el mapa del sitio.</p> <p>1.50. Cómo notificar a <i>Google</i> que el contenido del sitio <i>web</i> ha sido cambiado.</p> <p>1.51. Cómo instalar e implementar el <i>framework</i> para sitios múltiples.</p> <p>1.52. Cómo asignar contenidos a sitios múltiples.</p> <p>1.53. Cómo utilizar Django en un servidor Apache.</p> <p>1.54. Cómo optimizar un sitio <i>web Django</i>.</p> <p>2. Construir una aplicación básica para resolver el problema real seleccionado en la unidad 1, empleando <i>DJANGO</i>.</p>
--	---

Unidad 3: *SPRING*

Competencia específica a desarrollar	Actividades de Aprendizaje
Construir aplicaciones <i>web</i> elementales, haciendo uso de <i>SPRING</i> , siguiendo los principios y paradigmas en que se basa, para resolver problemas típicos del ámbito empresarial o del propio ambiente de <i>Internet</i> .	<p>1. Elaborar un resumen que incluya la definición de los siguientes conceptos:</p> <p>1.1. Inversión de control</p> <p>1.2. Inyección de dependencia</p> <p>1.3. Acoplamiento bajo</p> <p>1.4. Principio de no lo repitas tu mismo</p> <p>1.5. Contenedores de objetos</p> <p>1.6. Contexto de la aplicación</p> <p>1.7. Objetos perezosos</p> <p>1.8. Ámbito de los objetos</p> <p>1.9. Programación orientada a aspectos (AOP)</p> <p>1.10. Plantilla de base de datos</p> <p>1.11. Mapeadores relacionales de objetos (ORMs)</p> <p>1.12. Transacciones no invasivas</p> <p>1.13. Principio de responsabilidad única</p> <p>1.14. Propiedades de las transacciones (atomicidad, consistencia, aislamiento,</p>

	<p>durabilidad)</p> <p>1.15. Plantilla de transacción</p> <p>1.16. <i>Framework Acegi Security</i></p> <p>1.17. Autenticación</p> <p>1.18. Confirmación</p> <p>1.19. Objetos remotos</p> <p>2. Construir una aplicación básica para resolver el problema real seleccionado en la unidad 1, empleando <i>SPRING</i>.</p>
--	---

11.- FUENTES DE INFORMACIÓN

Fuentes impresas

1. Hoekman, Robert, Jr; Spool, Jared; ***Web Anatomy, Interaction Design Frameworks that work***; New Riders; United States of America; 2010.
2. Holovaty, Adrian; Kaplan-Moss, Jacob; ***The Definitive Guide to Django, Web Development Done Right***; Springer-Verlag; United States of America; 2008.
3. Dayley, Brad; ***Sams Teach Yourself Django in 24 Hours***; Sams Publishing; United States of America; 2008.
4. Bennett, James; ***Practical Django Projects, Write better web applications faster, and learn, how to build up your own reusable code library***; Second Edition; Apress; United States of America; 2009.
5. Walls, Craig; ***Spring in Action***; Third Edition; Manning; United States of America; 2011.
6. Konda, Madhusudhan; ***Just Spring***; First Edition, O'Reilly; United States of America; 2011.
7. Lee Turnquist, Greg; ***Spring Python 1.1***; Packt Publishing; United States of America; 2010.
8. Moore, Dana; Budd, Raymond; Wright, William; ***Professional Python® Frameworks, Web 2.0 Programming with Django®, and TurboGears™***; Wiley Publishing; ; United States of America; 2007.

12.- PRÁCTICAS PROPUESTAS

Se recomienda que estas prácticas sean llevadas a cabo por equipos diferentes de alumnos, con la finalidad de dosificar el trabajo involucrado por la asignatura. La intención es brindar experiencias que faciliten la aplicación del *framework* estudiado y apoyen en los trabajos señalados en las actividades de aprendizaje.

- Elaborar una aplicación *web* para los clientes de un banco que posea las siguientes características:
 - ✓ Vistas desacopladas y capa controladora.
 - ✓ Uso de diferentes autenticaciones simples para garantizar el acceso a diversos tipos de usuarios.

- ✓ Empleo de una autenticación basada en roles que permita distinguir entre diferentes tipos de clientes.
- ✓ Aplicación de autorización personalizada para evitar que algún cliente pueda ver los datos de otros.
- ✓ Exportación de datos en una red confiable en un formato plano que sea legible para la máquina.
- ✓ Exportación de datos a usuarios externos que siga protocolos de seguridad establecidos para proveerlos solamente a clientes autenticados y autorizados.
- ✓ Remarcado de operaciones de múltiples pasos como transacciones atómicas.

Algunas restricciones para la aplicación son las siguientes:

- ✓ Un cliente puede abrir una cuenta nueva con un saldo de \$0.0.
 - ✓ Un cliente puede cerrar una cuenta que tenga un saldo de \$0.0.
 - ✓ La escritura de la apertura y cerrado de cuentas solo puede ser visible al cliente propietario o cualquier administrador.
 - ✓ Un cliente puede retirar una cantidad menor o igual al monto del saldo.
 - ✓ Un cliente puede depositar cualquier cantidad en una cuenta que exista y que sea suya.
 - ✓ Un cliente puede transferir de una cuenta suya a otra cuenta que también le pertenezca un monto que sea menor o igual al saldo de la cuenta Fuente.
 - ✓ Todos los retiros, depósitos y transferencias de una cuenta pueden ser vistos por el cliente propietario o por cualquier administrador.
 - ✓ Los registros estarán disponibles a través de un formato seguro máquina-máquina, que requiera credenciales válidas.
 - ✓ Las acciones de un administrador que está viendo registros serán realizadas separadamente. Los registros de estas acciones serán visibles a un supervisor.
- Elaborar una aplicación *web* para un administrador de contenidos.
 - Crear una aplicación *web* para un portal de búsqueda.
 - Crear una aplicación una red social.
 - Elaborar una aplicación para inscripciones de alumnos.