



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



FACULTAD
DE INGENIERÍA

Ingeniería del Software II

TRABAJO PRÁCTICO GUÍA

Profesor:

- Lic. Leandro Spadaro.

Año:

- 2025

Objetivos

- a) Ejercitarse de forma rápida la identificación de clases, atributos, responsabilidades y sus relaciones mediante enunciados cortos.
- b) Realizar pequeños ejercicios mediante arquitectura cliente – servidor respetando lo indicado en cada ejercicio junto con sus respectivas pruebas automáticas, utilizando la metodología de desarrollo que el grupo decida (RUP, AGIL, libre, etc), fomentando el espíritu de investigación de los grupos en diferentes tecnologías y arquitecturas basados en el análisis, diseño e implementación orientado a objeto.
- c) El conocimiento adquirido en cada uno de los ejercicios pueda ser aplicados a los Trabajos prácticos integradores propuestos para el cursado de la materia.

Forma de trabajo.

El estudio y aprendizaje para la resolución de los ejercicios será con metodología autoestudio - autodidacta, donde se establecerá como material de consulta para la resolución de los ejercicios una guía audiovisual conformada por videos almacenados en la plataforma Youtube. Además, los grupos de trabajo podrán utilizar el material de consulta que considere necesario para complementar el conocimiento requerido (libros, videos, inteligencia artificial, etc).

Equipos de trabajo.

Los equipos de trabajo serán los conformados al inicio del cursado de la materia.

Presentación del Trabajo Práctico.

El grupo deberá presentar los ejercicios codificados en la clase de práctica según el cronograma establecido para tal fin. La codificación deberá ser almacenada en un repositorio público.

Cronograma de presentación:

- Fecha presentación Ejercicio N°1: 02/10/2025
- Fecha presentación Ejercicio N°2: 09/10/2025
- Fecha presentación Ejercicio N°3 y N°4: 16/10/2025
- Fecha presentación Ejercicio N°5 y N°6: 23/10/2025
- Fecha presentación Ejercicio N°7: 30/10/2025

Al finalizar todos los ejercicios se deberá presentar un documento que haga referencia al trabajo práctico donde deberá figurar el enunciado del ejercicio junto con el repositorio que contiene la codificación del mismo.

Codificación de Ejercicios.

Para la codificación de los ejercicios en los que se solicita la misma, cada grupo podrá utilizar el lenguaje de programación que considere que cumple con las pautas necesarias para cumplir el objetivo solicitado. El lenguaje de referencia que se utilizará durante el cursado es Java, mediante su implementación con el Framework Spring Boot ya que dicho lenguaje fue incorporado en materias cursadas de forma previa, permitiendo que la curva de aprendizaje sea reducida.

Patrones de programación.

Los grupos de trabajo deberán aplicar en la resolución de los ejercicios indicado el conocimiento adquirido al momento de realizar la programación.

Condición de aprobación del trabajo práctico.

A continuación, se detallan las condiciones de aprobación del trabajo práctico:

- 1) El 70% de cada ejercicio realizado funcionando donde se contempla lo especificado en el punto anterior.
- 2) Participación en clases.
- 3) El 80% de la asistencia individual a clases de práctica de cada miembro del equipo.
- 4) El 80% de asistencia a clases de consulta de práctica de al menos uno (1) de los integrantes del grupo de trabajo.

PARTE N°2.

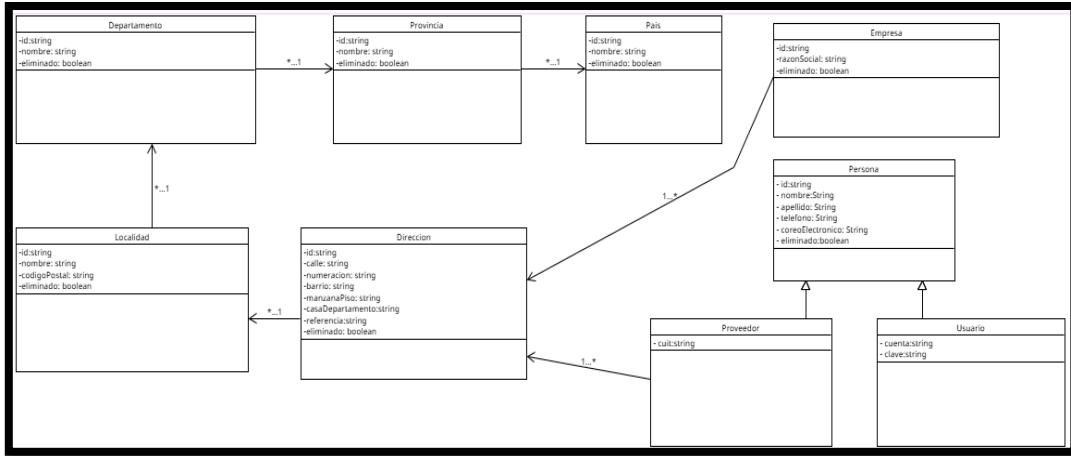
Ejercicio N°1: REPASO DE SCRUM / MVC

- a) Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondientes para siguiente problema planteado. El equipo fue contratado para desarrollar una red social, donde las personas que la utilicen tendrán un usuario para realizar publicaciones. Cada publicación contiene una imagen junto con un comentario. Los usuarios de la red social pueden visualizar las publicaciones de otros usuarios e indicar si les gusta, además de poder realizar comentarios. Los usuarios pueden enviar mensajes de forma privada entre ellos. El comportamiento de la red social es muy similar a la conocida “Instagram” y solo se ejecutará en Dispositivos Móviles del tipo celular.

Se deberá realizar las Historias de Usuario que contemplen las funcionalidades críticas especificadas junto con el prototipado de pantalla, y la estimación de cada historia de usuario utilizando Planning Poker. Investigar que tablero de seguimiento (Scrum / kanban) utilizaría el equipo para el desarrollo del proyecto.

- b) Realizar la codificación del FrontEnd y BackEnd que contemple el siguiente diagrama de clases. Deberán realizarse todos los ABM correspondientes para cada clase del diagrama.

Contemplar que los usuarios ingresan al sistema mediante “Login” con usuario y clave. Para el FrontEnd deberán buscar y utilizar una maqueta, gratuita, del tipo “Dashboard”, Ejemplo.: <https://startbootstrap.com/theme/sb-admin-2> .



El sistema debe tener menú, donde cada menú tendrá submenús de la siguiente forma:

1. Menú: Usuario.
 - a) SubMenú: ABM Usuarios.
 - b) SubMenú: Cambiar Clave.
2. Menú Dirección.
 - a) SubMenú ABM País.
 - b) SubMenú ABM Provincia.
 - c) SubMenú ABM Departamento.
 - d) SubMenú ABM Localidad.
3. Menú Empresa.
 - a) SubMenú ABM Empresa.
4. Menú Proveedor.
 - a) SubMenú ABM Proveedor.

Antes de comenzar el desarrollo se deberán realizar las historias de usuarios que contemplen las funcionalidades, junto con el prototipado de pantalla y las estimaciones correspondientes utilizando Planning Poker.

Finalizado el desarrollo coloque por cada Historia de Usuario el tiempo real que necesito el equipo para dejar la funcionalidad terminada.

Comprare el tiempo estimado con el tiempo real asumiendo que se encuentran en una reunión de Sprint Retrospective. Describa las conclusiones que obtuvo el equipo.

- c) Google Maps: Refactorice la clase dirección para que pueda tener los atributos latitud y longitud de tipo String. Estos atributos deben guardar la información de un punto de referencia de Google Maps. Ej.: -32.88970575178735 (latitud) -68.84457510855037 (longitud) (punto Plaza Independencia). Luego al mostrar la información de la empresa o de un proveedor se deberá modificar el Front para generar un link (botón) que abra en otra solapa del navegador el mapa con la ubicación. El link se debe formar con <https://www.google.com/maps?q=-32.88970575178735,-68.84457510855037>
- d) Automatización de Tareas: Realizar el ejercicio de la guía audio visual “[1\) AUTOMATIZACIÓN DE TAREAS y ARCHIVOS](#)”. Agregar código la posibilidad para enviar correos electrónicos de publicidad de la empresa, los correos se deben enviar los días 5 de cada mes a las 15 horas. Además, debe mandar correo a los proveedores cada 31 de

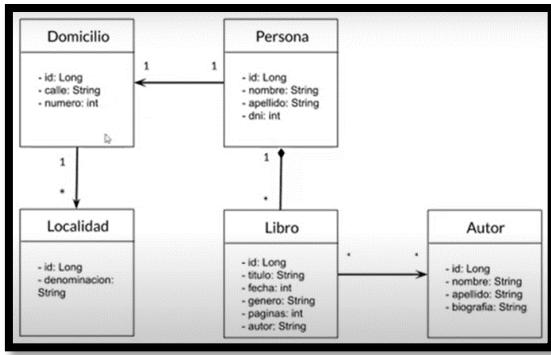
diciembre para realizar saludos de fin de año. El contenido de los correos deberán tener un HTML que contengan un botón que sea un link que al presionarlo se abra otra solapa en el navegador con la dirección URL de la facultad (<https://www.uncuyo.edu.ar/>).

- e) Archivos EXCEL y PDF: Realizar el ejercicio de la guía audio visual “**1) AUTOMATIZACIÓN DE TAREAS y ARCHIVOS**”. Agregar como funcionalidad al ejercicio “b” la posibilidad de descargar en archivo PDF el listado de Proveedores en el AMB Proveedor y descargar en Excel el listado de empresas en el AMB Empresa.
- f) Archivos TXT: Utilizando el ejercicio programado en el punto “b”, agregar el menú Migración, y realiza una funcionalidad que lea de un archivo llamado “migración.txt” el cual posee los datos de los proveedores con sus direcciones para ser ingresados en el sistema. El archivo posee todos sus campos separados por “;”, donde los campos son: NOMBRE;APELLIDO;TELEFONO;CORREO;ELECTRÓNICO;CUIT;CALLE;NÚMERO;LOCALIDAD;DEPARTAMENTO;PROVINCIA;PAIS. Realizar el ejercicio de la guía audio visual “**1) AUTOMATIZACIÓN DE TAREAS y ARCHIVOS**”. Ejemplo de la estructura interna del archivo:

Lionel;Escaloneta;2613654789;l.scalo@gmail.com;202934256790;Perú;345;Villa Hipodromo;Godoy Cruz;Mendoza;Argentina;
Pablo;Perez;26145622319;l.perez@gmail.com;20324326780;San Martín;1050;Capital;Mendoza;Argentina;
Esteban;Peralta;2615673490;l.estaban_p@gmail.com;20349874570;Benavente;121;Villa Nueva;Guaymallen;Mendoza;Argentina;

Ejercicio N°2: ARQUITECTURA API-REST.

- a) Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondiente para siguiente problema planteado: Se desea registrar en un sistema para Consultorios Médicos, los turnos de los pacientes. Los pacientes pagan un co-seguro para poder ser atendidos. El turno debe contener el horario en el que asiste el paciente y el médico que lo atenderá.
Previo a la realización del diagrama de clases, se deberá elaborar las Historias de Usuario que pudieran formar parte de la especificación, junto con sus prototipos y la estimación de tiempos utilizando Planning Poker. Además, se tiene que armar el Product Backlog del proyecto donde se registran las Historias de Usuario, uno de los miembros del equipo debe ejercer el rol de Product Owner para la priorización de las historias de usuario. Buscar al menos 3 Sitios Web del mercado que se dediquen al mismo rubro.
- b) Dado el siguiente modelo desarrolle el software de tipo web cliente – servidor que permita implementar las relaciones establecidas utilizando arquitectura REST mediante API. El proyecto deberá contar de dos aplicaciones una cliente y otro servidor que se comuniquen mediante API REST. La aplicación cliente deberá utilizar DTO's para manipular la vista luego de consumir las API del servidor para mostrarla en un sitio HTML.



- c) Crear un nuevo proyecto igual al desarrollado y agregar al punto anterior la posibilidad que al momento de crear un libro se pueda cargar el archivo PDF correspondiente en el disco C del servidor en la carpeta C://biblioteca//libro_nombrelibro_.pdf. Al momento de consultar el libro se debe poder abrir en una solapa distinta del sistema el PDF. Refactorizar el diagrama de clases y el código para cumplir con el objetivo.

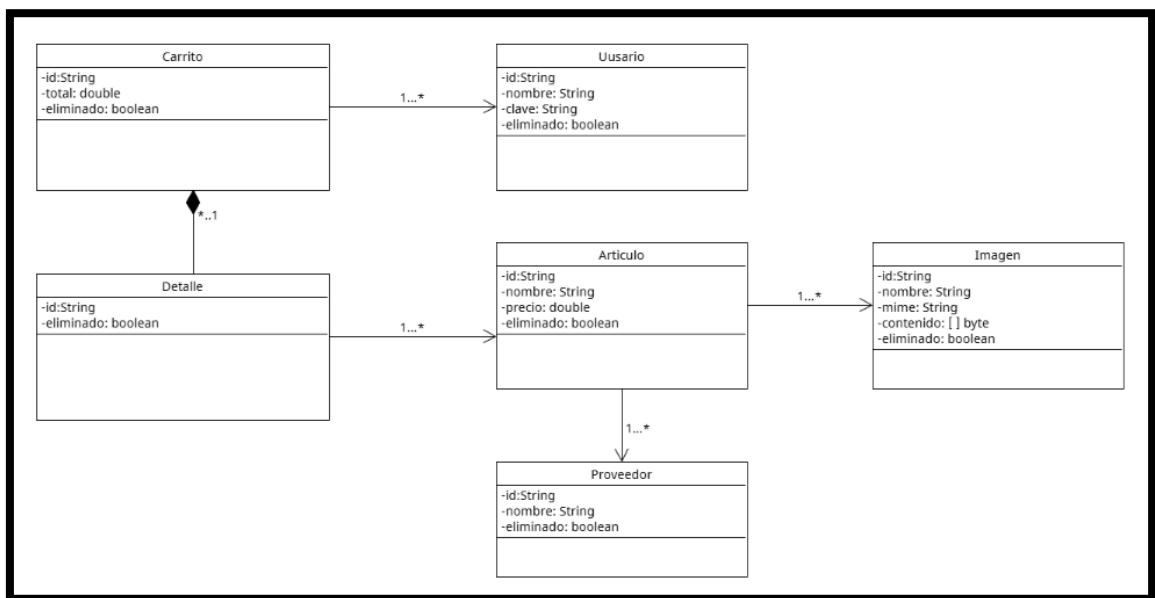
- d) APIS EXTERNAS: Realice un software que consuma información de una API Externa en el BackEnd, para luego ser visualizada en el FrontEnd.
 Utilizar sitios de internet para investigar sobre Apis externas.
 Ej.: <https://rapidapi.com/> o <https://openweathermap.org/api> o <https://magicloops.dev/es>
 Utilizar una maqueta gratuita HTML/CSS/Bootstrap para el FrontEnd.
 Ej.: <https://plantillashtmlgratis.com/categoría-plantillas/bootstrap/>
 Investigar el acceso la API de WhatsApp y la utilidad que el equipo podría darle en un proyecto futuro.

Ejercicio N°3: PATRÓN TEMPLATE METHOD / GENERIC / REFLECTION

- a) Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondiente para siguiente problema planteado: Trabajamos en un software que permite a los usuarios del mismos realizar el alquiler de cabañas en la provincia de Mendoza, donde cada cabaña puede tener distintos propietarios. Las cabañas tienen capacidad para 2, 4 o 6 personas. Las cabañas se alquilan con un mínimo de 5 días corridos.
 Realizar las Historias de Usuario críticas, junto con su estimación de tiempos utilizando Planning Poker y el presupuesto aproximado de costo mediante un documento formal, El equipo podrá utilizar como referencia los valores publicados en el sitio: <https://cpcipc.org.ar/honorarios-recomendados/>. Buscar al menos 3 Sitios Web del mercado que se dediquen al mismo rubro.

- b) El equipo ya adquiero experiencia en el análisis, diseño e implementación de software y observa que podría ahorrar tiempo en la codificación del modelo MVC y estudia la forma que puede utilizar en Patrón Plantilla (Template Method) en las clases Controlador y clases de Servicio. Para los equipos que utilice el lenguaje Java además podrá utilizar Java Generics y Java Reflection.
 El equipo utiliza todos los recursos para investigar Libros, Inteligencia Artificial, Ingeniería inversa de proyectos de años anteriores de la materia, etc.

- c) Crear un nuevo proyecto igual al desarrollado en el ejercicio 1 “b”, refactorice el código, para aplicar una clase controlador con Template Method y una clase de servicio con Template Method.
- d) Crear un nuevo proyecto igual al desarrollado en el ejercicio 2 “b”, refactorice el código, para aplicar una clase controlador con Template Method y una clase de servicio con Template Method.
- e) Realizar la codificación del siguiente diagrama de clases, aplicando el patrón Template Method, para un carrito de compras con su respectivo FrontEnd y BlackEnd teniendo en cuenta un Template gratuito para carrito de compras. Antes de comenzar el ejercicio estimar los tiempos de desarrollo con Planning Poker y comparar con los realmente ejecutados. Describir las conclusiones de los tiempos.

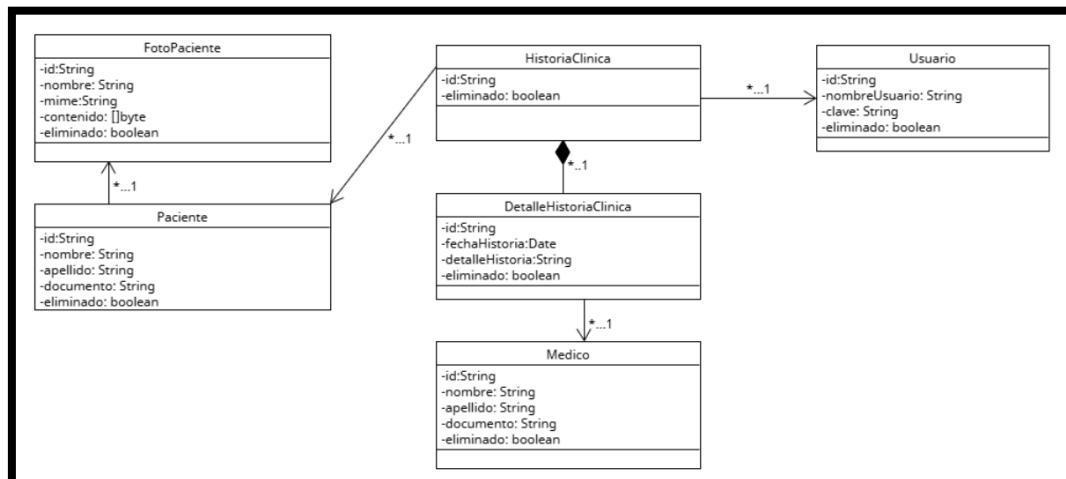


- f) OpenStreetMap: Investigar y agregar al sistema anterior la dirección de los proveedores donde se guarda la latitud y longitud de cada uno de ellos. El objetivo es mostrar en un mapa la ubicación de todos los proveedores con la librería indicada.

Ejercicio N°4: PATRONES CREACIONALES / ESTRUCTURALES / COMPORTAMIENTO.

- a) Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondiente del siguiente problema planteado: Contratan al equipo para desarrollar un software que permita registrar la compra de entrada para el cine de la universidad. Al momento de comprar una entrada para la película indicada se asigna el asiento y la sala. Cada entrada posee un valor que cambia cada determinado tiempo. Se deberán realizar las Historias de Usuario junto con el prototipado de pantallas, estimación de las Historias de Usuario con Planning Pocker, el Product BackLog con las priorizaciones de las Historias de Usuario. El presupuesto aproximado de costo mediante un documento formal, El equipo podrá utilizar como referencia los valores publicados en el sitio: <https://cpcipc.org.ar/honorarios-recomendados/> . Buscar al menos 3 Sitios Web del mercado que se dediquen al mismo rubro. Se incorpora los Sprint BackLog con las Historias de Usuario, cada Sprint tiene una duración de 2 semanas (5 días hábiles de 8 horas laborables).

- b) Crear un proyecto con el código del ejercicio Video Juego (5 “b” práctico guía Parte 1) y adaptarlo para utilizar API REST, aplicando el patrón Template Method.
- c) Se realiza la refactorización del ejercicio 3 “d” para incorporar patrones Creacionales, Estructurales, Comportamiento y Diseño:
1. Patrones Creacionales:
 - ✓ Builder: Utilizar el patrón para crear objetos del tipo Libro.
 - ✓ Fábrica: Utilizar el patrón para implementar la creación de Libros Físico y Libros Digitales. Refactorizar el código para que el sistema se adapte a la nueva funcionalidad.
 - ✓ Prototipo: Utilizar el patrón para realizar la clonación de un autor.
 2. Patrones Estructurales:
 - ✓ Adaptador: Utilizar el patrón para adaptar DTO en clase Entidad y viceversa.
 3. Patrones Comportamiento:
 - ✓ Estrategia: Utilizar el patrón para buscar libros por título, género y autor.
 - ✓ Iterador: Utilizar el patrón para recorrer una lista de libros y obtener los que son de un determinado autor.
- d) Reportes: Se le solicito al equipo que se realice una mejor para poder descargar en PDF la factura emitida en el sistema del “GIMNASIO SPORT”. El diseño del archivo que se descargue debe ser el similar a una factura de papel. Realizando la Guía Audio Visual el equipo adquiere el conocimiento para realizar la tarea designada. Ej.: formatos de factura: <https://invoicehome.com/?locale=es>
- g) Realizar la codificación del siguiente diagrama de clases (NO UTILIZAR API REST), con su respectivo FrontEnd y BackEnd, aplicando el patrón Template Method., referido a la historia clínica de un paciente que se atiende en un consultorio médico, teniendo en cuenta un Template gratuito. Antes de comenzar el ejercicio estimar los tiempos de desarrollo con Planning Poker y comparar con los realmente ejecutados. Describir las conclusiones de los tiempos.



Luego de realizada la codificación, proponga el uso de patrones creacionales, comportamiento y estructurales (Un patrón por categoría), refactorizando el código. En

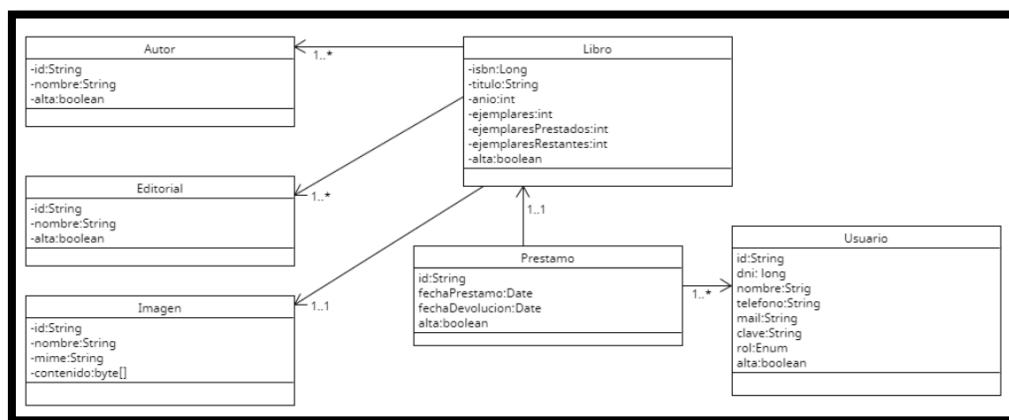
caso de ser necesario, el equipo, puede agregar nuevas clases o funcionalidades. Se debe fundamentar los motivos de los patrones utilizados explicando el concepto teórico y como se utilizan en el ejercicio.

Ejercicio N°5: SEGURIDAD.

- a) Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondiente del siguiente problema planteado: Trabajamos en el desarrollo de un software para colegios, donde se registran los alumnos con su respectivo grado y aula, junto a sus profesores. Además, se registran las notas por materias de los alumnos.

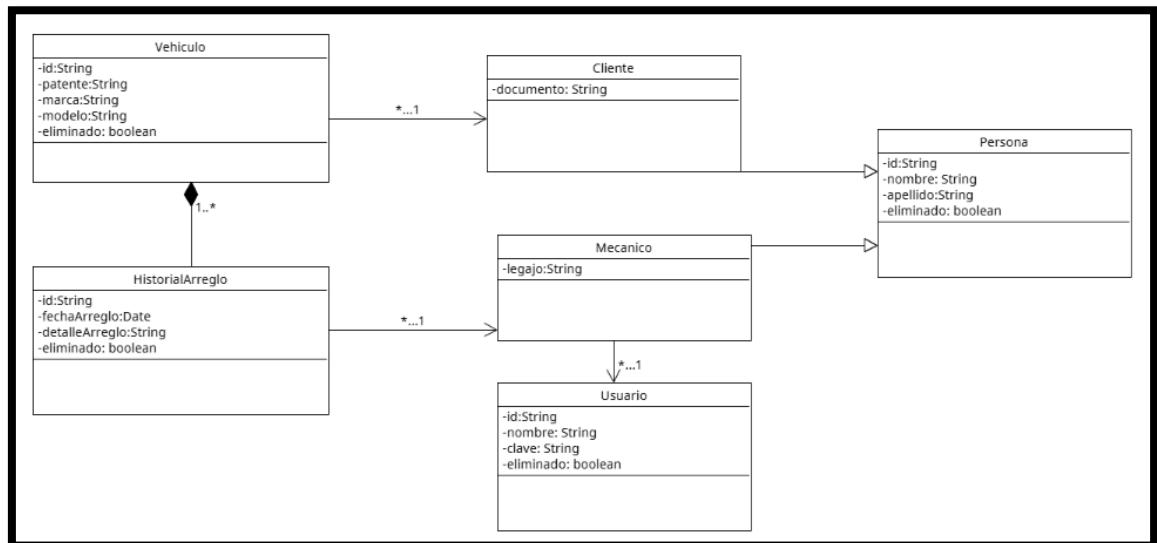
Realizar las Historias de Usuario, el prototipado de pantalla, la estimación de tiempos de las Historias. El presupuesto aproximado de costo mediante un documento formal, El equipo podrá utilizar como referencia los valores publicados en el sitio: <https://cpcipc.org.ar/honorarios-recomendados/>. Buscar al menos 3 Sitios Web del mercado que se dediquen al mismo rubro. Requisitos No funcionales.

- b) Dado el siguiente modelo desarrolle el software de tipo web que permita implementar las relaciones establecidas en el modelo aplicando conceptos de seguridad del FrameWork seleccionado por el equipo. Ver la Guía Audio Visual.



- c) Crear un nuevo proyecto igual al de “Mascotas” de la guía práctica N°1 y aplicar los conocimientos de seguridad adquiridos.
- d) Crear un proyecto igual al de “GYM SPORT” integrador N°1 y aplicar los conocimientos de seguridad adquiridos.
- e) Identifique los patrones de diseño utilizados por el Framework de seguridad seleccionado por el equipo.
- f) Refactorización de código:
1. Implementar Patrón Estrategia para realizar búsquedas de libros por año, editorial y autor.
 2. Implementar el Patrón Fachada para al momento de registrar un libro.
- h) Realizar la codificación del siguiente diagrama de clases (NO UTILIZAR API REST), con su respectivo FrontEnd y BackEnd, aplicando el patrón Template Method, referido a la

reparación de vehículos que se realizan en un taller mecánico, teniendo en cuenta un Template gratuito. Antes de comenzar el ejercicio estimar los tiempos de desarrollo con Planning Poker y comparar con los realmente ejecutados. Describir las conclusiones de los tiempos.



La aplicación debe contar con todas las medidas de seguridad aprendidas.

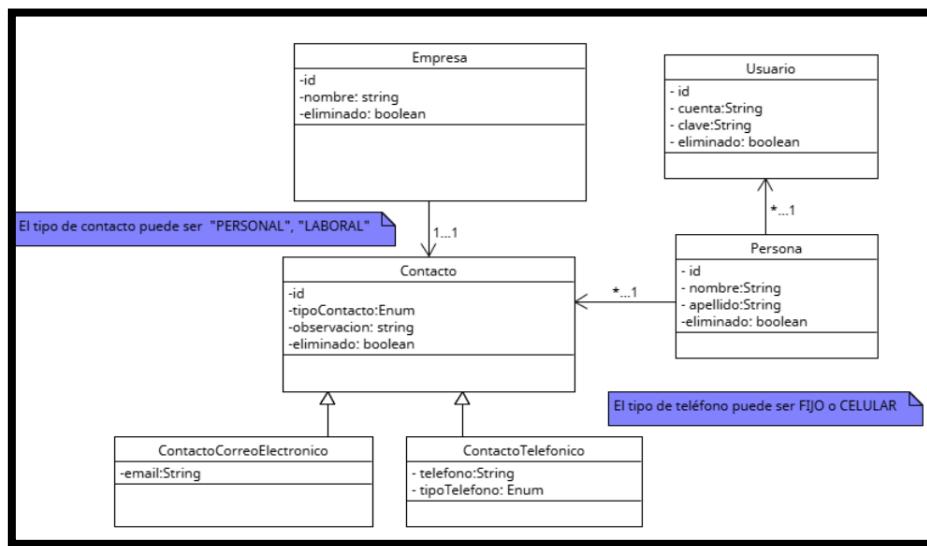
Luego de realizada la codificación, proponga el uso de patrones creacionales, comportamiento y estructurales (Un patrón por categoría), refactorizando el código. En caso de ser necesario, el equipo, puede agregar nuevas clases o funcionalidades. Se debe fundamentar los motivos de los patrones utilizados explicando el concepto teórico y como se utilizan en el ejercicio.

Ejercicio N°6: DESPLIEGUE.

- Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondiente del siguiente problema planteado: El equipo trabaja en el desarrollo de software para una cadena de restaurante (sucursal) donde cada restaurant posee una caja propia donde se registran los movimientos de ingreso y salida de dinero cada vez que un comensal realiza el pago de lo consumido.
Teniendo en cuenta que el Equipo accede a información confidencial, el cliente desea que se firme un acuerdo de confidencialidad. Realizar las Historias de Usuario, el prototipado de pantalla, la estimación de tiempos de las Historias. Realizar el presupuesto aproximado de costo mediante un documento formal, El equipo podrá utilizar como referencia los valores publicados en el sitio: <https://cpcipc.org.ar/honorarios-recomendados/>. Buscar al menos 3 Sitios Web del mercado que se dediquen al mismo rubro.
- Automatización de despliegue: Utilizando una herramienta de automatización de despliegue tipo DOCKER preparando la aplicación del equipo para despliegue en Producción. Visualizar los primeros 2do. videos de la guía “**6) Despliegue**”.
- Teniendo en cuenta la aplicación desarrollada en el ejercicio N°5 “d” – GIMNASIO SPORT con Seguridad, deberá implementar la puesta en marcha en producción en un servidor real. Para cumplir con el objetivo utilizará los servicios de despliegue ofrecidos por la

empresa de hosting “Don Web” <https://donweb.com/es-ar/> . Visualizar el 3er. video de la guía “**6) Despliegue**”.

- d) Integración Continua: N°5 “d” – GIMNASIO SPORT con Seguridad, para practicar el uso de servidor de integración continua, por ejemplo, para la automatización de despliegue o test unitarios. Visualizar el 4to. video de la guía “**6) Despliegue**”.
- i) Realizar la codificación del siguiente diagrama de clases (NO UTILIZAR API REST), con su respectivo FrontEnd y BackEnd, aplicando el patrón Template Method, referido a la registración de contactos de empresas y sus empleados, teniendo en cuenta un Template gratuito. Antes de comenzar el ejercicio estimar los tiempos de desarrollo con Planning Poker y comparar con los realmente ejecutados. Describir las conclusiones de los tiempos.



La aplicación debe contar con todas las medidas de seguridad aprendidas.

Luego de realizada la codificación, proponga el uso de patrones creacionales, comportamiento y estructurales (Un patrón por categoría), refactorizando el código. En caso de ser necesario, el equipo, puede agregar nuevas clases o funcionalidades. Se debe fundamentar los motivos de los patrones utilizados explicando el concepto teórico y como se utilizan en el ejercicio.

- e) WhatsApp Web: Realice una mejora en el Front, del ejercicio anterior, para que pueda ser posible abrir desde la aplicación, mediante un link, WhatsApp Web para el caso de los contactos de tipo celular. El link se debe formar con <https://wa.me/2613866379>

Ejercicio N°7: SEGURIDAD AVANZADO.

- a) Desarrollar el Diagrama de Clases de Diseño con sus atributos, relaciones, cardinalidades y métodos correspondiente del siguiente problema planteado: Trabajamos en un sistema que permite publicar noticias de la facultad. En la publicación se muestra una imagen, su contenido y el autor de la noticia junto con la fecha en la que fue escrita. Utilice el sitio de la facultad como referencia de investigación.
Realizar las Historias de Usuario, el prototipado de pantalla, la estimación de tiempos de las Historias. Realizar el presupuesto aproximado de costo mediante un documento

formal, El equipo podrá utilizar como referencia los valores publicados en el sitio: <https://cpcipc.org.ar/honorarios-recomendados/> . Buscar al menos 3 Sitios Web del mercado que se dediquen al mismo rubro.

El Equipo debe buscar maquetas html gratuitas que permitan adaptar los requerimientos del cliente con el objetivo de tener una aproximación real de los solicitado. Utilizar una maqueta con información real del sitio de la facultad.

Ej.: <https://plantillashtmlgratis.com/categoría-plantillas/educacion/>

- b) Crear un nuevo proyecto igual al de “Mascotas” de la guía práctica N°1 y aplicar los conocimientos de seguridad adquiridos.
- c) Crear un proyecto igual al de “GYM SPORT” integrador N°1 y aplicar los conocimientos de seguridad adquiridos.
- d) SEGURIDAD AUTH0: Generar un nuevo proyecto que contenga la misma implementación del ejercicio N°5 “b”. Al mismo se le deberá agregar seguridad AUTH0 para cliente (login) y servidor según corresponda.
- e) SEGURIDAD API REST: Generar un nuevo proyecto que contenga la misma implementación del ejercicio N°2 “b” y aplique seguridad, con el método que designe, para que la misma sea tenida en cuenta al momento de efectuar EndPoint.
- f) Teniendo en cuenta la aplicación desarrollada en el ejercicio N°7 “d” de Seguridad Avanzada, deberá implementar la puesta en marcha en producción en un servidor real. Para cumplir con el objetivo utilizará los servicios de despliegue ofrecidos por la empresa de hosting “Don Web” <https://donweb.com/es-ar/> . Visualizar el 3er. video de la guía “**6) Despliegue**”.

Mesas Examen Final.

Los alumnos que al finalizar el cursado de la materia su condición fuera de alumno regular, o en condición de alumno libre si la materia permitiera acceder a la instancia mencionada, para deberán concurrir el día de la evaluación con el trabajo práctico cumpliendo las siguientes condiciones:

- A) Alumnos Regulares y Alumnos Libres: Trabajo Práctico Completo e individual según lo especificado en cada punto.
- B) Alumnos Libres: Se deberá presentar cada sección “a)” de los ejercicios codificados teniendo en cuenta el Patrón MVC, arquitectura API-REST y aspectos Seguridad en el desarrollo. Se deberá contemplar utilizar patrones de diseño de software creacionales, estructurales y de comportamiento en cada uno de los ejercicios según el criterio del alumno, argumentando los motivos de la implementación de los mismos.

GUÍA AUDIOVISUAL

DE REFERENCIA

1) AUTOMATIZACIÓN DE TAREAS y ARCHIVOS

Desarrollo Ejercicio.

- Scheduling

<https://www.youtube.com/watch?v=nQBYCytWhJY>

- Archivos PDF

<https://www.youtube.com/watch?v= FUCwZ8Hf8w>

- Archivos Excel

https://www.youtube.com/watch?v=U_CDd796Lz0

<https://www.youtube.com/watch?v=azHQKsmrTil>

- Archivos de Text y tokens

https://www.youtube.com/watch?v=uqZEOO5MU_M&t=9s

<https://www.youtube.com/watch?v=ttgV5kfiH7Y>

2) API REST (HTTP + REST + API + JSON)

Conceptos Teóricos:

https://www.youtube.com/watch?si=a2cutnilG4rW_BP&v=RhxOTqFbI5Q&feature=youtu.be

<https://www.youtube.com/watch?v=yLG2pM5r3xs&list=PLgwlfcqa5h3xj-GyibK7yUPseOvl9puFm>

Desarrollo Ejercicio:

- API REST

<https://www.youtube.com/watch?v=LGck-So93TA&list=PLRFOqDrY-6nvhgKJ2Db1h8aPJm3zaX6eO>

- Subir Archivo en el Servidor

<https://www.youtube.com/watch?v=h5YW- xgmlo>

- API EXTERNA

<https://www.youtube.com/watch?v=LaJtCXROTvo>

<https://www.youtube.com/watch?v=oJ7-Ou9LcQ8>

Ejercicio Año 2024:

https://github.com/Coppede-Santos/Ing_software_2_tp_3

https://github.com/JacoR7/IS2_TP6_APIREST_CONSUMER

<https://github.com/aidalaricchia2/ApiRestConsumer> y <https://github.com/aidalaricchia2/ApiRest>

3 y 4) PATRONES DE DISEÑO.

Conceptos Teóricos:

<https://refactoring.guru/es/design-patterns>

https://www.youtube.com/watch?v=x-K6gDI5ZqA&list=PL4bT56Uw3S4x1_IhP0kQZ1mszuLvoNMJF

<https://medium.com/@abhishekranjandev/building-your-own-geolocation-platform-like-google-maps-using-openstreetmap-and-java-spring-boot-ec4ea6c77338>

<https://medium.com/@hermanmaleiane/spring-boot-thymeleaf-leaflet-js-mapping-corona-virus-a8309c5a0b6d>

Desarrollo Ejercicio:

- Reportes

<https://www.youtube.com/watch?v=pX7K0M7CTMU>

<https://www.youtube.com/watch?v=Gvwa- eNW5Y>

<https://www.youtube.com/watch?v=VxbWjeHXhTY>

5) Seguridad Parte 1 (Spring Security + OAUT2 + JWT)

Conceptos Teóricos:

<https://www.youtube.com/watch?v=qjPh0yrDNas>

<https://www.thymeleaf.org/doc/articles/springsecurity.html>

<https://www.baeldung.com/spring-security-thymeleaf>

<https://dev.to/chittrmahto/spring-boot-login-register-logout-with-mysql-thymeleaf-bootstrap-ui-show-user-details-33fl>

Desarrollo Ejercicio:

- Biblioteca

<https://www.youtube.com/watch?v=TXg0NGnxMaY&list=PLgwlfcqah3xL7coUYDfa8Ff6Z0n5iUt>

- Un poco más de seguridad

https://www.youtube.com/watch?v=aeCotM2DORo&list=PLr23_YfwEbPRCK4lbemQGwYdgSwfd2aZu

- Patrones en Seguridad:

https://www.youtube.com/watch?v=om5XZOsTQ_c

Ejercicio Año 2024:

<https://github.com/TomasRandoM/ingenieriaDelSoftware/tree/main/Ejercicio7/biblioteca>

<https://github.com/MartinFarres/biblioteca-Security>

<https://github.com/JacoR7/EjercicioSeguridad/tree/master/biblioteca>

6) Despliegue.

Desarrollo Ejercicio:

- DOCKER

https://www.youtube.com/watch?v=OEvhEEeDpg_o

<https://www.youtube.com/watch?v=cWuirzM0wwg>

- DEPLOY

<https://www.youtube.com/watch?v=cMXTd6PoFpo>

- JENKINS

<https://www.youtube.com/playlist?list=PLRFOqDrY-6ns9LOJn1XU0oOICnu8up26x>

7) Seguridad Parte 2 (Spring Security + OAUT2 + JWT)

Desarrollo Ejercicio:

- OAuth 2 y JWT con Spring Security

<https://www.youtube.com/watch?v=zDzvuTBrP1I&list=PLb9gJ6hvBcdon6FkVS5oavn6vQy134zv>

- Spring Security 6 + JWT

<https://www.youtube.com/watch?v=nwqQYCM4YT8>

Ejercicio Año 2024:

<https://github.com/paulisuden/SeguridadOAuth2>

<https://github.com/MauroSorbello/IngSoft2/tree/master/tp7/biblioteca/biblioteca>

<https://github.com/adrzanbar/rest-api-security>