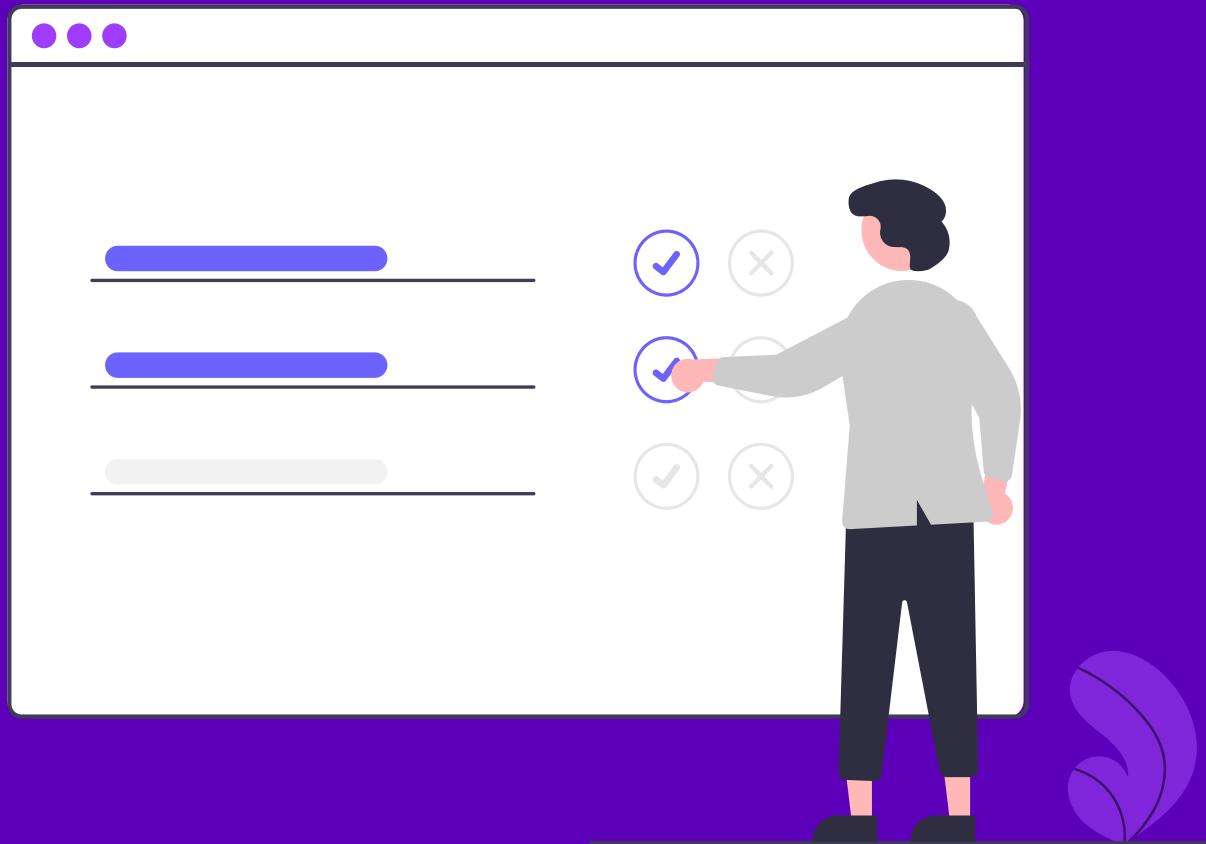


PLAN DE ESTUDIOS

# JAVA



### CLASE 1

#### Presentación y herramientas

- Java - Introducción y un poco de Historia
  - J2EE
  - Framework Spring Boot
- Eclipse: Versiones y plataformas compatibles
- Utilidades básicas de Eclipse (Tipos de proyectos, Debugger, Excepciones)
- Control de versiones utilizando Git y Github

### CLASE 2

#### Programación orientada a objetos y conceptos básicos

- POO
  - Encapsulamiento y accesibilidad de estado de objeto
  - Polimorfismo
  - Herencia
- Interfaces
- Alcance y tiempo de vida de variables
- C#
  - Tipos de datos genéricos
  - Clases parciales
  - Variables sin tipo de datos (var) y variables dinámicas
  - Extensión methods
  - Principios SOLID
  - Unit Test con JUnit y Jupiter

### CLASE 3

#### Inyección de dependencias

- Configuration Settings
  - Veremos la forma de parametrizar/configurar valores
  - Environment setting (Ej: Development, Staging, Production)
- Inyección de Dependencia
  - Cómo implementar el patrón de diseño Dependency Injection y cómo conectarnos a una base de datos usando Hibernate/JPA.
- Code First + Migrations

## CLASE 4

### Acceso a datos

- Operaciones CRUD
  - Como ejecutar las operaciones CRUD EF básicas
  - Ejecución diferida
- Lógica de Negocio y Patrón Repositorio
  - Cómo validar las reglas de negocio de nuestra aplicación, además profundizamos un poco en el Patrón Repositorio.
  - Generic Repository & Unit Of Work.

## CLASE 5

### Evaluación - Afianzar conocimientos desde clase 1 a clase 4.

- Exposición y revisión de código expuestos en clases anteriores.
- Evacuación de dudas de alumnos.
- Evaluación de alumnos a través de ejercicio práctico.

## CLASE 6

### HTTP, REST, RESTful

- Arquitectura Cliente-Servidor
- Http & REST
  - Fundamentos de HTTP, REST y Richardson Maturity Model
- Rest vs RestFull
  - Recursos
  - Request
  - Response
  - Payload / Body
  - Header (Accept / Content-type)
  - HTTP Verbs
  - HTTP Status codes
- CORS
- Servidor Web
- Postman

## CLASE 7

### Arquitecturas > Patrón MVC y otras arquitecturas

- MVC
  - Patrón de diseño
- Repaso básico de arquitecturas de software: MVVM, Onion, Hexagonal y microservicios
- Cómo crear la estructura de nuestra aplicación utilizando una Onion- MVC Architecture. Cómo decidir entre distintas arquitecturas

## CLASE 8

### Operaciones CRUD con Web API usando JMapper

- Dto & JMapper
  - Cómo evitar ataques de Overposting implementando la técnica de Data Transfer Object (DTO) y su correcta configuración utilizando JMapper.
- Filtering Data & Searching Data
  - Cómo implementar el filtro de datos usando el decorador [FromQuery] [FromBody] [FromService], etc.
- Negociación de contenido
  - Accept y Content-Type
  - XML y JSON
- Manejo de Excepciones
  - Respuestas HTTP según entorno (environment: Development/Production)

## CLASE 9

### Práctica - Afianzar conocimientos desde clase 6 a clase 8.

- Revisión y exposición de código fuente expuesto en clases anteriores.
- Evacuación de dudas de alumnos.
- Evaluación de alumnos a través de ejercicio práctico.

## CLASE 10

### Operaciones CRUD más avanzadas con Web API

- Operaciones CRUD y Typed Results
  - Cómo ejecutar las operaciones CRUD EF básicas para nuestra entidad POST y también cómo crear una respuesta genérica para nuestra RESTFul.
  - Tipos de retorno
- Patch para actualizaciones parciales (JsonPatchDocument)
- Validaciones
- Swagger
  - Cómo generar automáticamente la documentación de una API utilizando Swagger

## CLASE 11

### Seguridad/Autenticación y Deploy Web

- Repaso de conceptos básicos de redes
  - Dominios de red (intranet / extranet)
  - Domain Name Services (DNS)
  - Firewall
- Login de Usuario
  - Cómo registrar un nuevo usuario con el fin de poder generar un token de autenticación. Además vemos como realizar la validación del usuario contra la base de datos.
- Autenticación de usuarios con token
  - Cómo implementar una autenticación basada en tokens con JWT. Además vemos cómo invocar el API anexando un Bearer Token en el Header del request.
- Hashing Password
  - Como guardar las contraseñas de forma segura generando un HASH a partir de un Salt y una cantidad definida de iteraciones usando las funciones derivadas PBKDF2
- Deployar en IIS
  - Cómo desplegar nuestra aplicación tanto On-Premise como también en la Nube utilizando un hosting online.

### MÓDULO PRÁCTICO

Desarrollo de una aplicación tipo Blog, que cumpla con las siguientes funcionalidades:

- Un usuario anónimo (no registrado) podrá ver los posts pero no podrá editarlos ni borrarlos ni dejar comentarios.
- Un usuario anónimo podrá registrarse para convertirse en usuario registrado.
- Un usuario registrado podrá crear y editar sus posteos.
- Un usuario registrado podrá realizar un comentario sobre el posteo de otro usuario (No habrá comentarios de comentarios).
- Un usuario "Admin"/"Moderador" podrá eliminar cualquier post o comentario.

## CLASE 12

### Crud Artículos Persistente en BD

- Presentación del proyecto
- Se atienden dudas con alumnos
- Se da como tarea "CRUD de artículos persistidos en base de datos".
- BONUS TRACK: "Busqueda de articulos por tag"
- BONUS TRACK 2: "Búsqueda de artículos por palabra clave"

## CLASE 13

### Registro y Creacion de Usuario

- Entrega de tarea de "CRUD de artículos persistidos en base de datos"
- Se atienden dudas con alumnos
- Se da como tarea el inicio de "Registro y creación de usuario"
  - Creación del salt
  - Hashing de password
  - Persistencia en la base de datos
  - Verificación de existencia de usuario
- BONUS TRACK: "Paginación de Artículos"
  - Parametrizar tamaño de página y nro. de página deseada
  - Siempre debe retornar la cantidad total de artículos.

## CLASE 14

### Logueo de Usuario

- Entrega de tarea de “Registro y creación de usuario”
- Se atienden dudas con alumnos
- Se da tarea la “Logueo de Usuario”
  - Login de usuario
  - Generacion de Token
  - Patron Strategy
  - Validación de Password y Usuario
- **BONUS TRACK:** “Activación de cuenta por email”
  - Le llega un email al usuario para que active su usuario.
  - El usuario no se puede loguear hasta que activó su cuenta.
  - El link de activación debe tener un token/hash que expire luego de cierto tiempo.

## CLASE 15

### Integración Crud Artículos y Autenticación

- Entrega de tarea “Logueo de Usuario”
- Se atienden dudas con alumnos
- Se da como tarea “Integración CRUD y autenticación”
  - Security en endpoints
  - Validación de Create Edit y Delete de artículos por usuarios
- **BONUS TRACK:** “Like & Dislike”
  - Los usuarios pueden darle like o dislike a los artículos.
- **BONUS TRACK 2:** “Contador de likes”
  - Al obtener los comentarios, debemos conocer la cantidad de likes/dislikes que tiene hasta el momento.

## CLASE 16

### CRUD Comentarios

- Entrega de tarea de Integración CRUD y autenticación
- Se atienden dudas con alumnos
- Se da como tarea “CRUD de Comentarios + integración de comentarios en el response de los artículos”
  - Creación de comentarios sobre un Artículo
  - Mantener las relaciones en el comentario y el artículo.
- **BONUS TRACK:** Poder “arrobar” usuarios en los comentarios (Ej: @username).
  - Los usuarios deben ser válidos, existentes en la base de datos.
  - No puede “arrobarse” el mismo usuario dos veces en el mismo comentario.

## CLASE 17

### Planning y Diseño

- Entrega de tarea de “CRUD de Comentarios + integración de comentarios en el response de los artículos”
- Se atienden duda con alumnos
- Se da como tarea la integración de Swagger
- **BONUS TRACK:** “Crear Hilos de Comentarios”
  - Comentar un comentario
- **BONUS TRACK:** “Aregar negociación de contenido”
  - Poder enviar el header “accept” y que el response sea en formato JSON o XML según lo solicitado.

## CLASE 18

### Evacuación de dudas y refinamiento

- Entrega de tarea “Integración de Swagger”.
- Se atienden dudas con alumnos
- Cierre de Proyecto

## **CLASE 19**

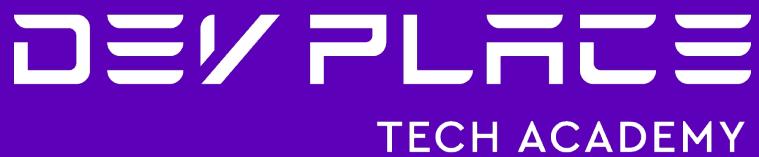
### **Evacuación Proyecto final**

- Evaluación de correcciones de alumnos que no hayan llegado a aprobar el proyecto final.

## **CLASE 20**

### **Evacuación Proyecto final**

- Evaluación de correcciones de alumnos que no hayan llegado a aprobar el proyecto final.



Contáctanos:

---

[aplicantes@devplace.com.ar](mailto:aplicantes@devplace.com.ar)