



UNIVERSIDAD
CAECE

Trabajo Integrador

Arquitectura Web

Dr. Miguel Méndez-Garabetti, Ing. Eduardo Piray

30/10/2025

Contenido

1.	Introducción.....	3
2.	Consideraciones iniciales.....	4
3.	Primera parte del Trabajo Integrador.....	6
3.1.	Actividad 1: aplicación web.....	6
3.1.1.	Requisitos para la aplicación web.....	6
3.2.	Actividad 2: consumir una API abierta.....	8
2.1.1.	Requisitos para la aplicación web de consumo.....	10
4.	Segunda parte del Trabajo Integrador: autenticación y ciberseguridad.....	12
5.	Anexo Matriz de Mitre.....	13
5.1.	Tácticas y técnicas.....	13
5.2.	¿Cómo utilizar la matriz de Mitre en el trabajo integrador?.....	14
6.	Entregables y modalidad de examen.....	15

1. Introducción

En el presente documento se definen los lineamientos para realizar la actividad integradora para regularizar/promocionar la asignatura Arquitectura Web. Se plantean los lineamientos a seguir para el desarrollo de aplicaciones web, en las cuales se deberán aplicar los conocimientos adquiridos durante el cursado de la materia, los cuales estuvieron relacionados al desarrollo de **RESTful API**.

2. Consideraciones iniciales

Forma de trabajo: grupal o individual. Los grupos pueden ser de hasta dos integrantes.

El trabajo final completo consta de dos partes:

- 1) **Primera parte:** Se plantean **dos ejercicios de desarrollo web**. Cada grupo o estudiante (en caso de trabajar de manera individual) deberán elegir uno de los ejercicios planteados y resolverlo. El primero de los ejercicios es una aplicación con un alcance idéntico a la primera aplicación hecha en clase. Mientras que el segundo ejercicio plantea el **consumo de una API Externa**, tal como se presentó en clase consumiendo la API de Pokémones.
- 2) **Segunda Parte:** Crear un **nuevo proyecto de Angular** y darle funcionalidad con una capa de seguridad, vinculada con la autenticación e identificación de al menos uno o dos usuarios.

También se plantea de forma opcional, añadir la misma capa de seguridad a alguna de las aplicaciones web hechas en la primera parte.

Base teórico-práctica desarrollo web: Para realizar los ejercicios, es necesario tener una base de conceptos y metodologías de implementación relacionados al framework de front end Angular y del lenguaje de programación TypeScript. Todos estos conceptos se analizaron durante el cursado de la asignatura. Las bases necesarias son

- Nociones de JavaScript y HTML.
- Uso y manipulación de datos en formato JSON.
- Manipular bases de datos en la nube. Más precisamente peticiones GET al servicio de Realtime Database de Firebase.
- Instalar y configurar entorno local para el framework Angular.
- Uso y manipulación de Postman.
- Peticiones HTTP de tipo GET.
- Comandos básicos de Angular CLI.
- Estructura de un proyecto de Angular.
- Componentes, directivas.
- Flujo de control de Angular (@If,@for).
- Interpolación de string y binding entre componentes.
- Comunicación entre componentes.
- Base introductoria en TypeScript (tipos de datos, inyección de dependencias). Traspilación.
- Manejo de rutas.
- Uso de servicios.
- Observables y programación reactiva.

- Despliegue de aplicaciones front end en proveedores de hosting. Más precisamente en el proveedor Netlify o similar.

Base teórico-práctica para la aplicación de seguridad en aplicaciones web: Es necesario armar una base de tecnologías que puedan brindarnos autenticación de usuario y seguridad a los usuarios en nuestra aplicación web.

- Autenticación con la plataforma WSO2.
- Contrastar la seguridad de la metodología de autenticación elegida con la matriz de Mitre ATT&CK.

3. Primera parte del Trabajo Integrador

Recordar que tienen que optar por la Actividad 1 o la Actividad 2.

3.1. Actividad 1: aplicación web

Desarrollar una **aplicación web de tipo SPA (Single Page Application)** que siga los lineamientos básicos de una RESTFull API, subirla al proveedor de hosting Netlify y que la misma sea accesible vía navegadores web. La elección de la funcionalidad de la aplicación web queda a criterio del estudiante (por ejemplo landing page, portafolio, sitio de e-commerce, sitio de facturación, entre otros).

3.1.1. Requisitos para la aplicación web

Frontend: se deberá desarrollar en el framework de front end Angular, incluyendo HTML y CSS básicos.

Para comenzar hacer lo siguiente:

- Crear un proyecto desde cero con Angular CLI en su computadora local.

Concretamente en esta sección de la aplicación los requisitos son:

- Página de inicio con un encabezado, un cuerpo, y un footer o pie de página. Colocar en el encabezado o en alguna sección de la página, un botón que indique login.
- El cuerpo de la página es el que tiene que variar según las transiciones entre componentes.
- Hojas de estilos globales: se puede utilizar algunos estilos que se trabajaron en clase, también se puede aplicar Bootstrap. O crear una personalizada según necesidad.
- Hojas de estilo de componentes (opcional): queda a criterio del estudiante si las implementa o no según su necesidad.

- Deberán implementar cinco (5) componentes.
 - Un componente para el encabezado,
 - Un componente para el pie de página.
 - Un componente para simular un login con un formulario. A este componente se accederá desde un botón ubicado en el encabezado o alguna sección de la página que el estudiante considere pertinente. La simulación de login consiste en un formulario con dos campos, el primero para cargar el nombre de usuario y el segundo para el email del usuario, al pulsar en el botón de envío, por medio de **señales en Angular** se debe mostrar un mensaje en consola o en otra sección de la página del mismo componente que diga “Hola emialUsuario Login exitoso” e imprimir también el objeto con los datos cargados en el formulario. Los campos del formulario deben limpiarse una vez se envíen los datos.
 - Los dos restantes componentes deberán implementar alguna funcionalidad diferente cada uno según el tipo de página web que se pretenda desarrollar.
- La transición entre componentes debe realizarse siguiendo la idea de los sitios SPA, por medio de rutas definidas en la clase app.routes.ts utilizando la directiva RouterLink.
- Deberán realizar, al menos, 3 peticiones GET al back end.
- Deberán implementar un servicio, en donde se efectúen las peticiones de tipo GET (OPCIONAL).
- Deberán utilizar el nuevo flujo de control con @If y @For.
- Deberán utilizar interpolación de string, binding de propiedades y de eventos.
- Deberán realizar una sola comunicación entre dos componentes usando INPUT y OUTPUT (OPCIONAL).
- En caso de necesidad (OPCIONAL) pueden manejarse con colecciones de datos tipo arreglo.

Backend: para este caso el estudiante puede optar por una las siguientes opciones:

- Back end con un árbol JSON definido en la carpeta asset del propio proyecto de Angular. Un JSON local podríamos decir.
- Back end con un árbol JSON definido en el proveedor de servicios de Firebase, concretamente utilizar el servicio **Realtime Database**. Para ello será necesario crear un proyecto en Firebase. Tal como se trabajó en clase.

Despliegue: La aplicación deberá estar desplegada en un servicio tal como Netlify o similar.

3.2. Actividad 2: consumir una API abierta

Consumir desde una aplicación frontend en Angular una las siguientes API abiertas, o pueden optar por una que consideren apropiada para su proyecto.

Se listan 13 APIs pasibles de ser utilizadas si lo consideran pertinente. Puedan usar una de ellas, tomar alguna idea, o usar alguna otra API que no esté en la lista o elegir una API del repositorio: <https://publicapis.io/>

1. JSONPlaceholder

- URL: <https://jsonplaceholder.typicode.com/>
- Descripción: La API de prueba por excelencia. Simula una API REST completa con endpoints para usuarios, posts, comentarios, álbumes y fotos. No requiere autenticación.
- Idea para el Proyecto: Crear una "Red Social de Práctica" o un "Blog". Desarrollar un frontend en Angular para listar usuarios, ver sus posts, agregar nuevos posts (simulados) y comentarios. Es útil para practicar CRUD, servicios y enrutamiento.

2. The Cat API

- URL: <https://thecatapi.com/>
- Descripción: Una API completa con miles de imágenes y datos sobre gatos. Permite buscar, filtrar por categorías y votar.
- Idea para el Proyecto: Desarrollar una "Galería de Gatos" o un "Buscador de Razas". Usar Angular para crear una interfaz que muestre imágenes, permita filtrar por raza (consumiendo el endpoint de razas) y guardar favoritos en el LocalStorage o en una base de datos sencilla de Firebase.

3. The Movie Database (TMDb)

- URL: <https://www.themoviedb.org/documentation/api>
- Descripción: Una base de datos masiva sobre películas y series de TV. Ofrece información detallada, reparto, trailers y imágenes. Requiere registrarse para obtener una API Key (gratuita).
- Idea para el Proyecto: Construir un "Buscador de Películas y Series". Ideal para practicar enrutamiento (navegar a una vista de detalles de una película), componentes reutilizables (para tarjetas de películas) y el manejo de API Keys. Se puede extender con una sección de favoritos usando Firebase Firestore.

4. OpenWeatherMap

- URL: <https://openweathermap.org/api>
- Descripción: Proporciona datos meteorológicos actuales, pronósticos y datos históricos para cualquier ubicación. Requiere API Key gratuita.
- Idea para el Proyecto: Crear una "Aplicación del Clima". Los estudiantes pueden practicar el consumo de APIs, el manejo de respuestas JSON complejas y la geolocalización del navegador para obtener el clima automáticamente. Se puede mejorar mostrando el pronóstico extendido con gráficos usando CSS y JavaScript.

5. REST Countries

- URL: <https://restcountries.com/>
- Descripción: Una API sencilla que ofrece información estructural sobre países, como capital, bandera, población, moneda e idiomas.
- Idea para el Proyecto: Desarrollar un "Wiki de Países" o un "Juego de Geografía". Es excelente para practicar la visualización de datos en tablas o tarjetas, implementar un buscador robusto y filtros (por continente, idioma). Se puede conectar con OpenWeatherMap para mostrar también el clima de la capital.

6. NASA APIs (APOD)

- URL: <https://api.nasa.gov/> (Ejemplo: Astronomy Picture of the Day - APOD)
- Descripción: La NASA ofrece varias APIs. La APOD (Astronomy Picture of the Day) devuelve la imagen o video astronómico del día con una explicación. Requiere una API Key gratuita.
- Idea para el Proyecto: Crear una "Galería Astronómica Diaria". Practicar el manejo de diferentes tipos de medios (imagen vs. video) y crear una interfaz que muestre no solo la del día, sino un archivo de días anteriores.

7. Unsplash API

- URL: <https://unsplash.com/developers>
- Descripción: Una de las bibliotecas de fotos de stock más grandes del mundo. Ofrece una API muy completa para buscar y mostrar imágenes. Tiene un plan gratuito generoso para desarrolladores.
- Idea para el Proyecto: Construir un "Buscador de Imágenes" al estilo Pinterest. Perfecto para trabajar con grids responsivos en CSS, búsquedas en tiempo real.

8. CoinGecko API

- URL: <https://www.coingecko.com/en/api>
- Descripción: Proporciona datos exhaustivos sobre criptomonedas, incluyendo precios, capitalización de mercado, volúmenes de trading y más. No requiere API Key para el plan gratuito básico.
- Idea para el Proyecto: Desarrollar un "Tracker de Criptomonedas". Crear un dashboard que liste las principales criptos, muestre gráficos de precios (usando una librería como Chart.js) y permita a los usuarios crear una "cartera virtual" para seguir sus inversiones, guardando los datos en Firebase.

9. JokeAPI

- URL: <https://v2.jokeapi.dev/>
- Descripción: Una API simple que devuelve chistes. Permite filtrar por categoría, tipo (de una o dos partes) y buscar por palabras clave.
- Idea para el Proyecto: Crear una "Máquina de Chistes". Es un proyecto más sencillo pero excelente para practicar el manejo de estados en la aplicación (cargando, éxito, error) y la personalización de la experiencia de usuario (e.g., "solo chistes de programación"). Se puede integrar con un sistema de favoritos.

12. Rick and Morty API

URL: <https://rickandmortyapi.com/>

- Descripción: Contiene información sobre cientos de personajes, imágenes, ubicaciones y episodios de la serie Rick and Morty. Es muy completa y bien estructurada.
- Idea para el Proyecto: Aunque es de una serie popular, es menos usada en ejemplos básicos que la PokeAPI. Se puede construir un "Wiki de Personajes" con paginación (la API lo soporta nativamente), filtros por estado (vivo, muerto), especie, etc. Es excelente para aprender a manejar datos paginados en Angular.

13. API de Marvel: oficial de Marvel. <https://developer.marvel.com>

Algunas APIs requieren la creación de una cuenta, y también obtener una API key (llave para la API) para usarlas.

2.1.1. Requisitos para la aplicación web de consumo

La aplicación deberá tener implementado:

Frontend:

- Página de inicio con un encabezado, un cuerpo, y un footer o pie de página. Colocar en el encabezado o en alguna sección de la página, un botón que indique login.
- El cuerpo de la página es el que tiene que variar según las transiciones entre componentes.
- Hojas de estilos globales: se pueden aplicar estilos según necesidad.
- Hojas de estilo de componentes (opcional): queda a criterio del estudiante si las implementa o no según su necesidad.
- Deberán implementar cuatro (4) componentes.
 - Un componente para el encabezado,
 - Un componente para el pie de página.
 - Los dos restantes componentes deberán implementar alguna funcionalidad diferente de acuerdo a peticiones hechas a la API abierta elegida.
- La transición entre componentes debe realizarse siguiendo la idea de los sitios SPA, por medio de rutas definidas en app.module.ts y utilizando la directiva RouterLink.

- Deberán realizar, al menos, 4 peticiones GET a endpoints de la API abierta elegida.
- Deberán implementar un servicio, en donde se efectúen las peticiones de tipo GET antes mencionadas (OPCIONAL).
- Hacer las peticiones GET desde un backend en Express (ESTA CONSIGNA ES OPCIONAL).
- Se puede consumir cualquier tipo de información que la API devuelva en JSON: imágenes, nombres de personajes, información de personajes en general, y todo lo que el estudiante considere pertinente.

Backend: será la propia API abierta. Leer la documentación que provee cada API en los enlaces indicados con anterioridad.

Despliegue: La aplicación deberá estar desplegada en un servicio tal como Netlify o similar si ha trabajado sin backend. Si optó por trabajar con un backend en Express puede completar esta actividad trabajando en localhost, no es necesario un despliegue en hosting, nube o backend as a service.

4. Segunda parte del Trabajo Integrador: autenticación y ciberseguridad

Crear un nuevo proyecto de Angular y darle funcionalidad con una capa de seguridad, vinculada con la autenticación e identificación de al menos uno o dos usuarios utilizando la plataforma de **WSO2**. También se plantea de forma opcional, añadir la misma capa de seguridad a alguna de las aplicaciones web hechas en la primera parte de la actividad integradora.

1. Explorar la matriz de Mitre ATT&CK (disponible en <https://attack.mitre.org/>) para familiarizarse con las técnicas, tácticas y procedimientos de ataque y defensa más habituales en ciberseguridad.
2. Explorar e indagar en los protocolos para el diseño de sistemas de autenticación y sistemas de autorización.

Se recomienda indagar en los siguientes protocolos:

- 2.1. OAuth.
- 2.2. OpenIdConnect.
3. Analizar las metodologías de autenticación que podrían integrarse a la aplicación web que hayan desarrollado, o al proyecto web nuevo. Para ello deberán explorar la plataforma WSO2.
4. Identificar posibles vulnerabilidades que podría tener la aplicación web y los eventuales método de autenticación elegidos. Relevar dicha información en base a la matriz de Mitre ATT&CK (disponible en <https://attack.mitre.org/>).
5. Integrar autenticación a la aplicación web creado o alguna de las ya desarrolladas, utilizando los servicios de WSO2. Ayudarse con la clase sincrónica grabada.

5. Anexo Matriz de Mitre

MITRE introdujo ATT&CK (Adversarial Tactics, Techniques & Common Knowledge) en 2013 como una forma de describir y categorizar los comportamientos de los adversarios basándose en observaciones del mundo real. ATT&CK es una lista estructurada de comportamientos conocidos de los atacantes que han sido recopilados en **tácticas** y **técnicas** y expresados en un puñado de **matrices**. Dado que esta lista es una representación bastante completa de los comportamientos que los atacantes emplean cuando comprometen las redes, es útil para una variedad de medidas ofensivas y defensivas, representaciones y otros mecanismos.

MITRE ATT&CK es el resultado de un experimento de MITRE que emulaba tanto al atacante como al defensor para ayudar a comprender cómo se producen los ataques y mejorar la detección posterior a la vulneración usando telemetría y técnicas de análisis del comportamiento. Para comprender mejor la capacidad del sector a la hora de detectar comportamientos enemigos documentados, crearon el marco de trabajo ATT&CK como herramienta para clasificar estos comportamientos.

MITRE ha dividido ATT&CK en varias matrices diferentes: **Enterprise**, **Mobile** e **ICS Industrial Control Systems**. Cada una de estas matrices contiene diversas tácticas y técnicas asociadas con el tema de esa matriz.

1. La matriz Enterprise se compone de técnicas y tácticas que se aplican a los sistemas Windows, Linux y/o MacOS.
2. Mobile contiene tácticas y técnicas que se aplican a los dispositivos móviles: Android e iOS.
3. ICS contiene tácticas y técnicas relacionadas con los ambientes industriales.

5.1. Tácticas y técnicas

Cuando se observa ATT&CK en forma de matriz, los títulos de las columnas de la parte superior son **tácticas** y, esencialmente, categorías de **técnicas**. Las tácticas son lo que los atacantes están tratando de lograr (resultados u objetivos deseados). En las filas se representan las técnicas utilizadas para lograr los objetivos tácticos, es decir, son la forma en que logran objetivos.

Por ejemplo, una de las tácticas es el **Movimiento Lateral**. Para que un atacante logre con éxito el movimiento lateral en una red, querrá emplear una o más de las técnicas enumeradas en la columna de Movimiento Lateral en la matriz ATT&CK.

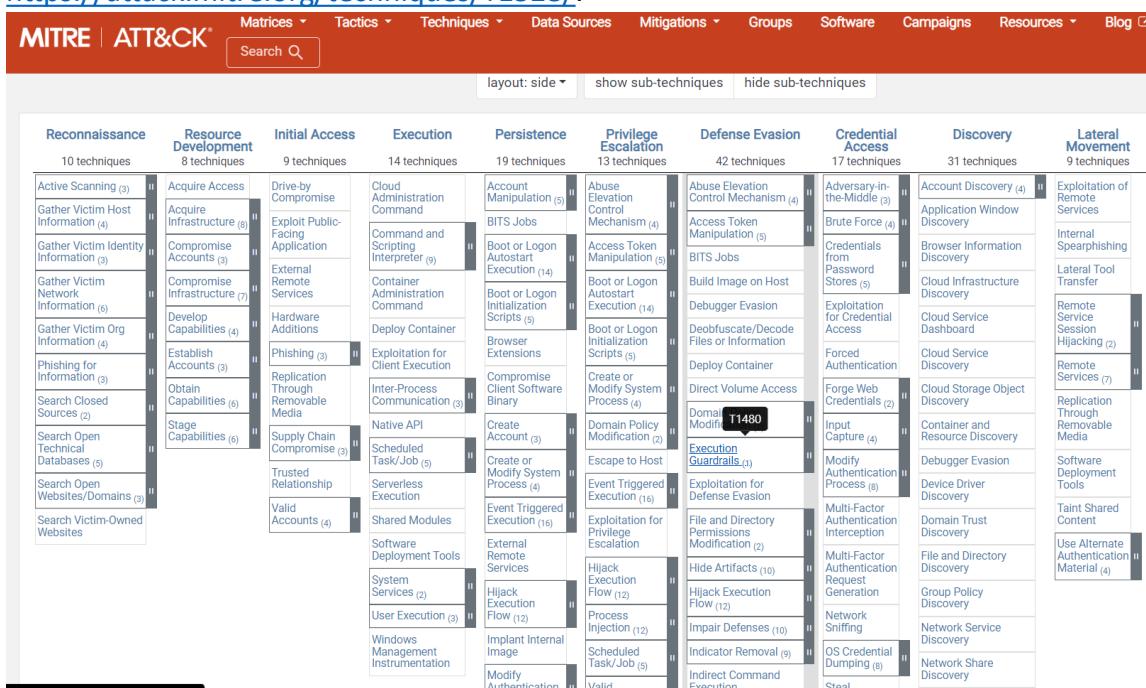
Una técnica es un comportamiento específico para lograr un objetivo y es a menudo un solo paso en una cadena de actividades empleadas para completar la misión general del atacante.

ATT&CK proporciona muchos detalles sobre cada técnica, incluyendo una descripción, ejemplos, referencias y sugerencias para su mitigación y detección.

5.2. ¿Cómo utilizar la matriz de Mitre en el trabajo integrador?

Para trabajar con este marco se recomienda:

1. Analizar las posibles vulnerabilidades de seguridad que pudieran presentarse en el sistema de autenticación elegido, listarlas y detallarlas. Las vulnerabilidades analizarlas en base a la matriz ofrecida en <https://attack.mitre.org/>, explorando tácticas y técnicas. Por ejemplo, analizar las tácticas de la matriz vinculadas credenciales de acceso. Para ser un poco más específico podríamos hacer esto, si la aplicación usa OAuth para autenticación, esta podría ser vulnerable al robo del token de acceso a la aplicación: <https://attack.mitre.org/techniques/T1528/>.



2. Analizar planes de mitigaciones a las posibles vulnerabilidades que hayamos identificado en base a la matriz de Mitre. Cuando se ingresa al detalle de cada técnica en la matriz, se pueden observar las posibles mitigaciones.

6. Entregables y modalidad de examen

Actividad	Entregable	Defensa del proyecto
Actividad 1	<p>1) Documento PDF de 8 páginas con observaciones y opiniones del proyecto realizado. Incluir una sección completa al documento referida a como se trabajó la ciberseguridad.</p> <p>2) El sitio web debe estar desplegado en un proveedor de hosting tal como Netlify o similar.</p>	Durante un espacio por videoconferencia el día de la fecha de examen asignada. Cada grupo deberá exponer su proyecto operando en la web, explicar el código implementado compartiendo pantalla de su ordenador local.
Actividad 2	<p>1) Documento Word de 8 páginas con observaciones y opiniones del proyecto realizado. Incluir una sección completa al documento referida a como se trabajó la ciberseguridad.</p> <p>2) El sitio web debe estar desplegado en un proveedor de hosting tal como Netlify o similar.</p>	Durante un espacio por videoconferencia el día de la fecha de examen asignada. Cada grupo deberá exponer su proyecto operando en la web, explicar el código implementado compartiendo pantalla de su ordenador local.

Cronograma de defensas

Grupo	Integrantes	Fecha de defensa oral
1		13/11/25
2		13/11/25
3		13/11/25
4		20/11/25
5		20/11/25
6		20/11/25
7		20/11/25