

## Introducción

El presente documento tiene como objetivo presentar el caso de estudio a emplearse como proyecto final de la carrera de Técnico en Redes y Software para el grupo TA3. Se buscará durante este tercer semestre desarrollar e implementar la solución que responda a las necesidades planteadas, cumpliendo con estándares de calidad, performance y seguridad teniendo aspectos fundamentales como alta disponibilidad, balance de carga, tolerancia a fallos y distribución geográfica de los servicios esenciales. El proyecto se realizará en supervisión de los tutores englobando las destrezas y conocimientos adquiridos a lo largo de toda la carrera.

## Reglamentación

Los alumnos formarán equipos para la realización del mismo.

La asistencia es obligatoria el primer mes, luego del primer mes los equipos pueden asistir cada 2 semanas, no siendo necesaria la asistencia de la totalidad del grupo de proyecto.

La aprobación del semestre requiere cumplir con los requerimientos planteados por los tutores, los cuales no abarcan el 100% del proyecto.

La calificación final del semestre es la misma para todos los integrantes de un equipo, y en caso de ser de suficiencia.

Para que quede habilitada la entrega final, ningún miembro del equipo debe contar con asignaturas previas del primer y segundo semestre. El equipo tiene 2 años a partir de la fecha de reunión final y entrega de boletines para realizar solicitar la entrega y defensa final, independientemente si tienen previas que los inhabilitan o no. La entrega final no puede realizarse en menos de 6 meses (un semestre) de haber aprobado la tutoría.

Si uno o más miembros de un equipo se encuentran inhabilitados por previas para realizar la entrega final, el resto del equipo puede optar por realizarla sin él. De esa forma, los miembros restantes pueden realizar la entrega final cuando queden libres de previas y cuenten con habilitación para hacerlo.

Dentro de los 2 años que cuenta cada equipo para realizar la entrega final, se tiene 2 oportunidades para hacerlo. Una vez vencido el plazo de 2 años, o que el juicio de 2 entregas no sea de "Aprobado", el equipo deberá recurrir el semestre de tutoría, realizando un proyecto nuevo.

La aprobación de la entrega final de proyecto consta de 4 partes:

- Entrega de documentación en la escuela
- Instalación de proyecto en la escuela
- Presentación final
- Defensa final

Cada parte es eliminatoria, y fallar en una implica automáticamente en un fallo negativo, y realizar otra entrega con correcciones, si se cuenta con otra oportunidad, o la realización de un nuevo proyecto en un semestre de tutoría si no se cuenta con más oportunidades.

El proyecto cuenta con los siguientes hitos:

<b>Primer Hito: Mayo</b>	<b>Segundo Hito: Julio</b>	<b>Tercer Hito: Pre-defensa final</b>
Presentación de propuesta de solución de cada grupo	Entrega de avance	Entrega final
<p>Resultado: presentación grupal para todos los tutores</p> <p>Los tutores brindarán feedback y se terminarán de definir las funcionalidades</p>	Resultado: entrega de carpeta digital intermedia	Resultado: entrega de carpeta final y presentación grupal

## Caso de Estudio

Se desea desarrollar un sistema de E-Commerce.

El sistema debe permitir que cualquier usuario (sea una persona física o un comercio) pueda registrarse para publicar artículos a vender, así como comprar los productos.

Además, se desea contar con funcionalidades de coordinar entrega de productos, así como que los cadetes puedan hacer seguimiento de qué productos tienen para entregar.

## Generalidades del caso

### Aplicación de Usuario

Esta aplicación permitirá visualizar los productos disponibles para comprar, así como realizar ventas

Debe permitir el registro de usuarios, los cuales pueden ser personas físicas, así como comercios. Si un usuario no está registrado, igualmente debe tener acceso de lectura a la aplicación.

La aplicación NO se encarga de administrar los productos que venden los usuarios, en su lugar, debe redirigir a los usuarios a la aplicación correspondiente.

Debe enviar notificaciones a los usuarios, sean promociones genéricas (por ejemplo, descuentos), respuestas a preguntas en las publicaciones, o aspectos relacionados a productos que tiene publicados a la venta (preguntas que le realicen, ventas concretadas, etc).

Las empresas que lo deseen, pueden crear un perfil personalizado llamado "Tienda". En la tienda, además de listar los productos que tiene disponibles, debe poder mostrar su logotipo, realizar personalización básica de colores (no del 100% de la página).

Se debe implementar utilizando un framework de Frontend, de forma que se comunique con APIs para acceder a la información, y el contenido estático de la aplicación debe implementarse usando un CDN (sea creado propio, o externo)

### Aplicación de Gestión

Esta aplicación es donde se realiza la gestión de los productos a vender, independientemente del tipo de usuario.

En esta aplicación se realiza el ABML de productos a vender, así como otros aspectos relacionados, como preguntas que le realicen los usuarios, así como personalizar el perfil de tienda, si es un usuario de tipo empresa.

Esta aplicación, por más de que sea enfocada a la gestión, es de visualización pública, por lo que se debe tener en cuenta el diseño de una interfaz acorde, y que tenga relación con el diseño de la aplicación de usuario.

Se debe implementar utilizando un framework de Frontend, de forma que se comunique con APIs para acceder a la información, y el contenido estático de la aplicación debe implementarse usando un CDN (sea creado propio, o externo)

## Aplicación de Delivery

Esta aplicación se encarga de que otro tipo de usuario, los cadetes, puedan ver que paquetes de compras tenga para enviar, los cuales se asignan dependiendo de la zona donde se encuentra el vendedor, así como marcar si pudieron entregarlo o no.

Deben poder visualizar, además, la ubicación en GPS de donde se encuentra la dirección del cliente para realizar la entrega.

Debe desarrollarse como dos aplicaciones (frontend y Backend).

## API de Productos

Esta aplicación es una API pública que brinda los productos disponibles y toda su información para ser visualizada y manipulada en las aplicaciones públicas.

La API debe ser RESTFUL.

## API de Autenticación

Esta aplicación es una simple API pública que se encarga de gestionar la autenticación. Se le brindan credenciales y retorna una respuesta JSON indicando si las credenciales son correctas o no. Solo es usada por las aplicaciones públicas.

La API debe ser RESTFUL, y debe desarrollarse de forma que funcione como un proveedor de autenticación (ej. OAuth), o comunicarse con un sistema externo (otro proveedor de OAuth, Active Directory o LDAP). A su vez, debe soportar autenticación mediante Facebook, Google y Twitter.

Sea como sea la implementación a usar, debe contar con su propia base de datos.

## API de Tiendas

Esta aplicación es una API pública que da información y gestiona los datos de las tiendas de los usuarios de tipo empresa.

La API debe ser RESTFUL.

## Backoffice de Administración

Esta aplicación de acceso restringido se encarga de gestionar el funcionamiento interno de todo el sistema (usuarios, publicaciones, cadetes, entregas, compras, así como aspectos de configuración general de la plataforma). Puede desarrollarse como dos aplicaciones (frontend y Backend), o como una aplicación monolítica clásica.

## Implementación

Todas las aplicaciones deben estar “dockerizadas” para ejecutarse en Kubernetes. La interacción de las aplicaciones con otros servicios de la infraestructura debe estar definida de tal forma que la persistencia no se de en los contenedores, y que sea posible la escalabilidad de las aplicaciones.

Se debe tener en cuenta la necesidad de servicios de red para el funcionamiento del sistema, como Base de Datos, Directorio Activo, Caché, Correo electrónico, almacenamiento de repositorios Git, DHCP, DNS, los cuales deben ser implementados.

## Ambientes

Se deben tener en cuenta 3 ambientes para desarrollar el proyecto, los cual deben definir y aclarar sus particularidades:

- **Test:** Este ambiente es donde los alumnos desarrollarán el proyecto. Está compuesto por sus PCs personales, en las cuales desarrollarán el software, y donde crearán máquinas virtuales para probar la infraestructura
- **Stage:** Es el ambiente donde se realizará la entrega final del proyecto. Se ejecutarán los servidores y las aplicaciones en equipos de la escuela. Todos los componentes del proyecto deben estar funcionales
- **Producción:** Es el ambiente donde correría el software y la infraestructura en su implementación final. No se llegará a implementar, pero debe diseñarse una propuesta de implementación utilizando proveedores de cloud computing, así como se debe entregar un presupuesto de la solución.

## Objetivos

- Diseñar y desarrollar todas las aplicaciones que componen el sistema
- Diseñar e implementar la solución de infraestructura de red para el funcionamiento del sistema
- Diseñar e implementar la solución de infraestructura de red para las oficinas de la empresa, la cual consiste en un modelo que se repetirá para cada oficina. Solo debe tener servicios esenciales para el funcionamiento de una oficina.
- La aplicación web deberá funcionar desde cualquier navegador y ser lo más intuitiva y fácil de usar por el público destinatario

## Puntualizaciones

- El sistema debe tener tolerancia a fallos (a nivel de datos y de operatividad)
- La infraestructura debe implementar interoperabilidad entre plataformas Microsoft y UNIX
- La configuración de los servidores debe implementarse y entregarse mediante un software de aprovisionamiento de configuraciones (Ansible, Puppet, Chef, etc)
- La autenticación de las aplicaciones debe ser implementada con algún servicio ya existente para dicho fin (oAuth2, LDAP, Active Directory). No se acepta implementar la autenticación y seguridad de forma manual mediante tablas.
- La infraestructura debe implementar soluciones de monitoreo y alertas.
- Tanto la infraestructura como las aplicaciones deben manejar los logs generados de forma centralizada.
- Se debe proveer un sistema de respaldo integral, para todos los servidores.
- La infraestructura para el ambiente de producción debe estar diseñada e implementada de forma que se adapte al uso de tecnologías en la nube. Salvo en el caso de las oficinas, no se acepta implementar como modelo un

Data Center local para los servicios.

- Las aplicaciones desarrolladas deben ejecutarse en contenedores Docker.
- El código de las aplicaciones debe estar versionado utilizando Git, usando un proveedor de hosting de repositorios (GitHub, GitLab, Bitbucket)
- También, mediante Git, se debe implementar Integración y Entrega continuos.
- El componente de Frontend debe desarrollarse utilizando JavaScript y HTML, así como cualquier framework que trabaje con ellos (Angular, ReactJS, VueJS, por ejemplo).  
El componente de Backend debe estar desarrollado en PHP, así como cualquier framework que lo utilice (Laravel, CodeIgniter, por ejemplo).

Para ambos casos se debe justificar su elección.

Quedan excluidos frameworks, paquetes o herramientas generadoras de código y lenguajes de 4to nivel como GeneXus

- El motor de base de datos debe ser MySQL o SQL Server, justificando su elección.
- Cada funcionalidad de cada aplicación debe tener asociadas pruebas unitarias o de integración que demuestren su correcto funcionamiento. Las mismas deben realizarse a medida que se realiza el desarrollo, no posteriormente.

El proyecto deberá contar con la debida documentación que corresponda a las distintas etapas del ciclo de vida del proyecto.

Además, se deberá desarrollar una guía de uso en línea enfocada al usuario final, que sea lo más interactiva posible.

## Equipos de Trabajo

**El trabajo será realizado en grupos de hasta 4 estudiantes.** Grupos de más integrantes se analizarán como excepciones. No se aceptarán trabajos individuales.

**El grupo deberá haber presentado antes del día 14/04/23 la conformación de los grupos,** vía correo electrónico a los docentes.

## Pautas para el trabajo

Dado a la naturaleza terciaria y técnica de la carrera, se entiende que las tareas realizadas por el proyecto deben tener el nivel de completitud correspondiente al nivel de la carrera.

El trabajo en el proyecto exige la integración de todos los conocimientos adquiridos en todas las asignaturas en los dos primeros semestres así como la adquisición de nuevos conocimientos por parte del grupo. El hecho de no existir tutorías correspondientes a asignaturas cursadas en el primer y segundo semestre (Diseño web, Conectividad, Seguridad de Redes, Auditoria de Redes, Ingeniería de Software e inglés), no exime la obligatoriedad de la aplicación de dichas asignaturas en el proyecto.

Es de especial importancia el contacto con cada uno de los docentes de las áreas temáticas del proyecto, a efectos de evacuar dudas, en las tutorías que corresponda en cada asignatura. Los docentes podrán solicitar vía correo electrónico cualquier información que estimen pertinente y tendrá la misma validez que hecha en clase.

El proyecto final que se entregue, debe poder ser ejecutado por los docentes en los equipos del Instituto. No se puede entregar ni instalar en máquinas personales de los alumnos.

## Entregas Semestrales

Las entregas semestrales se realizan de forma digital, vía correo electrónico a los docentes, para la documentación.

El código fuente de todos los aspectos del proyecto deben estar hospedados en un repositorio de Git utilizando alguna plataforma para este fin.

Los requerimientos puntuales de cada tutoría, y los criterios de evaluación de las entregas, se acordarán con cada tutor.

Las entregas se realizarán semestrales siguiendo el siguiente calendario:

**Primer Entrega: 5/05/2023**

**Segunda Entrega: 14/07/2023**



## Entrega final

Los requerimientos de la entrega final serán definidos por cada tutor. Cuando un grupo (o parte de un grupo) cumpla con la habilitación de entrega final por estar libre de previas desee realizar, deberá solicitar fecha a la escuela.

Los parámetros de entrega, instalación, presentación y defensa serán entregados por la escuela y tutores.