

EVALUACIÓN	Obligatorio	GRUPO	Todos	FECHA	01/09/2025
MATERIA	Arquitectura de Software en la práctica				
CARRERA	Ingeniería en Sistemas				
CONDICIONES	<p> - Puntaje máximo: 30 puntos - Puntaje mínimo: 15 puntos - Fecha de entrega: 09/10/25 hasta las 21:00 horas en gestion.ort.edu.uy (max. 40Mb en formato zip, rar o pdf) </p> <p>Uso de material de apoyo y/o consulta</p> <p><u>Inteligencia Artificial Generativa</u></p> <ul style="list-style-type: none"> - Seguir las pautas de los docentes: Se deben seguir las instrucciones específicas de los docentes sobre cómo utilizar la IA en cada curso. - Citar correctamente las fuentes y usos de IA: Siempre que se utilice una herramienta de IA para generar contenido, se debe citar adecuadamente la fuente y la forma en que se utilizó. - Verificar el contenido generado por la IA: No todo el contenido generado por la IA es correcto o preciso. Es esencial que los estudiantes verifiquen la información antes de usarla. - Ser responsables con el uso de la IA: Conocer los riesgos y desafíos, como la creación de “alucinaciones”, los peligros para la privacidad, las cuestiones de propiedad intelectual, los sesgos inherentes y la producción de contenido falso - En caso de existir dudas sobre la autoría, plagio o uso no atribuido de IAG, el docente tendrá la opción de convocar al equipo de obligatorio a una defensa específica e individual sobre el tema <p>Defensa</p> <p>Fecha de defensa: entre la semana 10 y 11 del curso</p> <p><u>La defensa es obligatoria y eliminatoria.</u> El docente es quien definirá y comunicará la modalidad, y mecánica de defensa. <u>La no presentación a la misma implica la pérdida de la totalidad de los puntos del Obligatorio.</u></p> <p>IMPORTANTE:</p> <ol style="list-style-type: none"> 1) Inscribirse 2) Formar grupos de hasta 3 personas del mismo dictado 3) Subir el trabajo a Gestión antes de la hora indicada (ver hoja al final del documento: “RECORDATORIO”) <p>Aquellos de ustedes que <u>presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos</u>, por favor contactarse con el Coordinador de cursos o Coordinación adjunta antes de las 20:00h del día de la entrega, a través de los mails croso@ort.edu.uy / posada_l@ort.edu.uy (matutino) / larrosa@ort.edu.uy (nocturno), o vía Ms Teams.</p>				

Objetivos del obligatorio

Desarrollar los conceptos que definen la computación en la nube mediante la adopción de técnicas de construcción de software como servicio, desde el desarrollo hasta su despliegue en un entorno de producción.

Introducción

Usted forma parte del equipo de tecnología de una startup en el espacio de la gestión empresarial. Entre una larga lista de proyectos, la empresa tiene como objetivo ofrecer una aplicación de gestión de relaciones con los clientes (CRM por su sigla en inglés) bajo el modelo SAAS.

Un sistema o software **CRM** es una herramienta que ayuda a centralizar y automatizar diversas funciones, incluyendo:

- **Gestión de contactos:** Almacena información detallada sobre clientes y prospectos (posibles clientes, también llamados “leads”).
- **Seguimiento de ventas:** Permite a los equipos de ventas seguir el progreso de los acuerdos, desde el primer contacto hasta el cierre de la venta. A este proceso también se le conoce como el embudo (funnel) de ventas.
- **Servicio al cliente:** Ayuda a gestionar solicitudes, incidencias y consultas de los clientes de manera eficiente.
- **Automatización de marketing:** Facilita la creación y el seguimiento de campañas de marketing, como el envío de correos electrónicos masivos.

Es su objetivo diseñar y construir el sistema según los requerimientos funcionales y no funcionales que se describen en este documento para lograr:

1. Entregarlo en el tiempo pactado.
2. Deslumbrar a los usuarios.
3. Establecer las bases de la arquitectura con escalabilidad y mantenibilidad en mente, pero sin perder de vista la realidad y el presupuesto actual.

Desarrollo

Para esta primera versión, su equipo ha determinado que una **arquitectura monolítica cloud native** es la más apropiada. Su equipo debe determinar el lenguaje, framework, tecnologías web y plataforma a utilizar para la construcción y despliegue de la aplicación, dada la situación de la empresa.

Requerimientos Funcionales

RF1. Registro de una empresa cliente

Actor: administrador

Los administradores del SAAS podrán registrar una empresa cliente, de la que se conocerá su nombre, dirección, sitio web (url) y logo, todos datos mandatorios. Cada empresa representa un cliente del SAAS y por lo tanto sus datos son privados. Bajo ningún concepto se debe compartir datos entre empresas.

Nota: en esta primera entrega los usuarios administradores serán definidos directamente en la base de datos.

RF2. Registro de usuarios de empresa mediante invitación

Actor: administrador

Un usuario administrador deberá poder invitar a usuarios miembros de una empresa a la plataforma, enviándole un correo electrónico de un solo uso (correo electrónico real, no simulado) que contenga un link a una página de ingreso, y con un mensaje que indique a qué empresa se está uniendo. Los usuarios siempre quedan definidos por su email y tienen asociado un password, que se le pedirá al acceder al link de invitación. Un usuario (email) solo puede pertenecer a una empresa y es de tipo miembro (no puede crear empresas ni invitar otros usuarios).

RF3. Autenticación de usuario

Actor: administrador y usuario miembro

El usuario debe poder acceder a un formulario de ingreso que solicita su correo electrónico y una contraseña de acceso. Una vez autenticado el usuario, el sistema debe redireccionarlo a la pantalla de inicio para la empresa de la que forma parte o en el caso de usuario administrador al dashboard de administrador. También es necesario que el usuario pueda cerrar sesión.

RF4. Gestión de claves de aplicación

Actor: usuario miembro

Se espera que la información de este sistema sea accesible desde otros sistemas mediante un api REST. Para esto, el sistema debe permitir a los usuarios miembros crear claves de acceso para otras aplicaciones (API KEYS). Un usuario miembro puede crear o dar de baja claves de aplicaciones (API KEYS) en cualquier momento. Al crearla deberá indicar un nombre, y el sistema devolverá la key, que solo será visible al ser creada. De allí en adelante no será posible ver el valor de la key.

RF5. CRUD de cuentas

Actor: usuario miembro

El sistema debe permitir la creación, edición y eliminación de perfiles de cuentas (empresas) con campos como nombre de la empresa (único), industria, dirección, número de empleados y sitio web. Estas cuentas representan clientes o potenciales clientes, y para cada una de ellas debe ser posible gestionar una lista de contactos, con campos nombre, cargo, email, teléfono y dirección.

Tanto para cuentas como para contactos se debe ofrecer funcionalidad de creación, modificación y eliminación. Las políticas a seguir en caso de eliminación (borrado lógico o en cascada, por ejemplo) serán decisión del equipo, que debe documentarlas. El único requerimiento es que la información se debe mantener consistente en todo momento.

RF6. Gestión de oportunidades

Actor: usuario miembro

El sistema debe permitir la creación y seguimiento de oportunidades de venta (deals) con atributos como cuenta, descripción, etapa del proceso de venta (Preparación, Presentación, Negociación, Venta, No Venta), valor estimado y fecha de cierre esperada. Estas oportunidades deben poder moverse a través del embudo o de ventas, reflejando las diferentes etapas del proceso.

Cada cambio de etapa debe quedar registrado con una fecha, el usuario que lo realizó y una descripción mandatoria de por qué se cambia de estado.

Debe ser posible adjuntar documentos (cotizaciones, propuestas, contratos, material presentado, etc.) a estas oportunidades de venta.

RF7. Página de inicio de empresa

Actor: usuario miembro

Al ingresar a la aplicación, los usuarios miembros deben ver una pantalla que permita ver un listado de todas las oportunidades abiertas (que no están en estado Venta / No Venta). El listado debe estar paginado (queda a elección del desarrollador de a cuantos, en función de lograr una buena visualización). Y debe permitir filtrar por cuenta, o cambiar el filtro por estado a cualquier combinación de estados.

Además del listado, se debe mostrar un gráfico que agrupe las oportunidades por estado.

RF8. Consulta de cuentas top (REST)

Actor: usuario miembro

Se deberá disponibilizar un endpoint REST que permita consultar las 3 cuentas con más oportunidades convertidas en ventas de la historia para la empresa. La request debe validar la API_KEY (RF4) para identificar a la empresa.

RF9. Consulta de cambios de etapa (REST)

Actor: usuario miembro

Se deberá disponibilizar un endpoint REST que dada una etapa, una fecha de inicio y una fecha de fin, devuelva todas las oportunidades que hayan pasado a esa etapa en el período. La request debe validar la API_KEY para identificar a la empresa.

Requerimientos No Funcionales

Usted y su equipo han relevado los siguientes requerimientos no funcionales para la etapa actual del proyecto:

RNF 1. Performance

Su equipo ha acordado que los endpoints públicos deben responder rápidamente para no perjudicar la performance de las aplicaciones que lo consumen. Los endpoints de los requerimientos funcionales relativos al api deben responder con una máxima de 300 ms para cargas de hasta 1200 req/m (p95).

RNF2. Confiabilidad y disponibilidad

Con el fin de poder monitorear la salud y disponibilidad del sistema, se deberá proveer un endpoint HTTP de acceso público que informe el correcto funcionamiento del sistema (conectividad con bases de datos, colas de mensajes, disponibilidad para recibir requests, etc.).

RNF3. Observabilidad

Para facilitar el diagnóstico ante eventuales fallas, usted y su equipo deberán poder monitorear al menos las siguientes métricas del sistema:

- Peticiones por minuto
- Tiempos de respuesta de distintos endpoints
- Tasa de error de los distintos endpoints

Debe ser posible consultar y visualizar esta información en cualquier momento.

RNF 4. Seguridad, autenticación, autorización y tenancy

Se deberá establecer un control de acceso basado en roles, distinguiendo entre usuarios administradores y usuarios miembros. Los permisos otorgados a los mismos deberán restringir el acceso a sus correspondientes funcionalidades, prohibiendo la interacción con cualquier otra operación pública del sistema.

Los usuarios de una empresa no pueden, bajo ningún concepto, acceder a datos de otra empresa.

Para los endpoints públicos, se sugiere que las claves de acceso a la aplicación sean generadas como JSON Web Tokens. El sistema debe validar que el token sea válido y tenga los permisos necesarios para acceder a todos los endpoints públicos.

El equipo debe determinar e implementar la mejor forma para asegurar que los datos sensibles de los usuarios sean protegidos como corresponde, incluyendo contraseñas, api keys, o cualquier otro dato considerado “secreto” de la aplicación. Las tácticas o medidas implementadas deben estar correctamente documentadas.

El sistema deberá responder adecuadamente en cualquiera de los casos de error descritos anteriormente, por ejemplo, mostrando una pantalla que describa el error en el front end y retornando

un código de estado descriptivo en los endpoints del back end. El sistema también deberá responder adecuadamente a cualquier URL malformada.

Por otra parte, la comunicación entre componentes de back end deberá realizarse dentro de una red de alcance privado. De lo contrario, deberán utilizar un protocolo de transporte seguro autenticado mediante el mecanismo que consideren necesario.

RNF 5. Código fuente

El control de versiones del código se deberá llevar a cabo con repositorios Git debidamente documentados, que contengan en el archivo README.md una descripción con el propósito y alcance del proyecto, así como instrucciones para configurar un nuevo ambiente de desarrollo.

Su equipo puede realizar un rebranding al nombre del sistema a su gusto. Para el manejo de branches, se deberá utilizar Gitflow o Github Flow y hacer uso de Pull Requests.

RNF 6. Integración continua

Se deberá contar con pruebas unitarias para al menos dos requerimientos funcionales de su elección. Estos dos requerimientos deben tener una cobertura de 100%. Las pruebas unitarias se deben correr automáticamente cada vez que un nuevo commit se integra a la rama principal.

RNF 7. Pruebas de carga

Se deberá diseñar y ejecutar un script de generación de planes de prueba de carga utilizando la herramienta k6, con el objetivo de demostrar el cumplimiento con los requerimientos de performance.

RNF 8. Identificación de fallas

Para facilitar la detección e identificación de fallas, se deberán centralizar y retener los logs emitidos por la aplicación en producción por un período mínimo de 24 horas.

RNF 9. Portabilidad

El repositorio debe contar con los archivos necesarios para que cualquier persona que se clone el repositorio y tenga Docker corriendo, pueda levantar el sistema de manera local con solo un comando.

Documentación

Además de la solución a implementar, se pide incluir una documentación con los siguientes puntos:

Descripción de la arquitectura

Deberá mostrar las partes que componen al sistema a través de vistas de módulos, componentes y conectores, y despliegue. Deberá asegurarse de mostrar:

- Distribución de todo el sistema en componentes físicos.
- Flujo de información a través de la infraestructura en tiempo de ejecución.

Justificaciones de diseño

Se pide una descripción del razonamiento y las tácticas aplicadas para conseguir cada uno de los RNFs exigidos. Las mismas pueden incluir tanto decisiones explícitamente introducidas en su diseño, como aquellas provistas por la plataforma, herramientas o framework utilizados. La documentación debe evidenciar el cumplimiento con cada uno de los RNFs, especialmente para aquellos que no sean verificables desde el código, como:

- RNF3. Observabilidad
- RNF4. Seguridad, autenticación, autorización y tenancy
- RNF8. Identificación de fallas.

Cumplimiento con 12 factor app

La documentación debe incluir una breve descripción (no es necesario más de un párrafo por factor) que evidencie o explique cómo la aplicación cumple con cada uno de los principios propuestos por [The Twelve-Factor App](#).

Descripción del proceso de deployment

Se deberá describir y justificar la plataforma utilizada para desplegar la app, así como los pasos tomados por su equipo para su configuración y/o despliegue manual (en forma de tutorial).

Además, se deberá mencionar brevemente el proceso interno que toma lugar al desplegar bajo el mecanismo descrito. Esto es, los pasos que comprenden el pasaje del sistema en su entorno de desarrollo al entorno de producción (compilación de assets, ejecución de pruebas, migraciones, etc.). Se debe explicar si el proceso utilizado ocasiona downtime al desplegar una nueva versión de la aplicación.

Equipos

Los equipos podrán estar conformados por hasta 3 personas, y pueden incluir integrantes de más de un dictado. No se podrá realizar el trabajo en forma individual.

Entrega

La entrega podrá realizarse a través de gestion.ort.edu.uy. La entrega debe incluir:

- URI para el acceso web público a una instancia en producción del sistema.
- Dirección de los repositorios privados en Github.
- Se debe incluir en alguno de los repositorios, el script de pruebas de cargas mencionadas en el RNF6 Integración continua.
- Link o adjunto de la Documentación requerida para este obligatorio.
- Colección de Postman para los dos endpoints REST públicos.
- Cualquier conjunto de credenciales adicionales que sean necesarias para el uso del sistema.

Datos de prueba

Se solicita que en el ambiente de producción se dejen ya registrados dos usuarios con datos de prueba para poder verificar la visualización de las gráficas y los valores históricos.

- Nombre empresa: ORT
- Usuario administrador: admin@saas.com Contraseña: Password1
- Usuario miembro: usuario@saas.com Contraseña: Password1

Defensa

La defensa se realizará mediante un video que cada equipo tiene que enviar a los docentes mostrando el funcionamiento de la aplicación. En el video se deberán mostrar todos los requerimientos funcionales implementados, y mencionar, en caso que haya, aquellos que no se hayan podido implementar.

Recomendamos revisar la siguiente rúbrica para validar que se cumple con lo pedido:

Requerimientos Funcionales	Items a validar
RF1. Registro de una empresa cliente	<ul style="list-style-type: none"> ✓ Permite registrar una nueva empresa cliente con los datos pedidos ✓ No permite registrar empresas con información faltante (nombre, dirección, sitio web y logo).
RF2. Registro de usuarios	<ul style="list-style-type: none"> ✓ Permite a administradores registrar nuevos usuarios para una empresa, incluyendo el envío del email de bienvenida. ✓ No permite el registro de usuarios sin los datos completos ✓ Restringe el registro de un usuario a una sola empresa. ✓ Obliga a los usuarios a resetear la contraseña después del primer login.
RF3. Autenticación de usuario	<ul style="list-style-type: none"> ✓ Permite el inicio de sesión estándar mediante email y password. ✓ Limita la visualización de datos a los pertenecientes a la empresa del usuario. ✓ Permite cerrar sesión. ✓ Impide el acceso a datos de otras empresas.
RF4. Gestión de claves de aplicación	<ul style="list-style-type: none"> ✓ Permite crear claves ✓ Se puede ver la lista de claves existentes, pero solo el nombre, no el valor ✓ Permite eliminar claves existentes.
RF5. CRUD de cuentas	<ul style="list-style-type: none"> ✓ Alta de cuentas, datos mandatorios y opcionales. ✓ Modificación de cuentas, manteniendo unicidad del nombre ✓ Baja de categorías. No deja datos inconsistentes.
RF6. Gestón de oportunidades	<ul style="list-style-type: none"> ✓ Alta de oportunidades con datos mandatorios, usuario de registro y fecha ✓ Cambios de estados con los datos pedidos
RF7. Página de inicio de empresa	<ul style="list-style-type: none"> ✓ Muestra los datos pedidos ✓ Pagina en forma consistente ✓ Filtra según los criterios especificados

RF8. API REST. Consulta de cuentas top	✓ Valida el apikey ✓ Toma la empresa del api key provisto ✓ Devuelve los datos pedidos
RF8. API REST. Consulta de cambios de etapa	✓ Valida el apikey ✓ Toma la empresa del api key provisto ✓ Valida los parámetros y los utiliza para filtrar ✓ Devuelve los datos pedidos

Los equipos podrán enviar el video de la defensa hasta una semana exacta después de la fecha de entrega del obligatorio.

Importante:


- Las funcionalidades que no se muestren en la defensa no se considerarán como implementadas.
- El video deberá contar con voz narrando lo que se va mostrando, puede durar como máximo 20 minutos.
- Durante la defensa se debe demostrar que todos los componentes del sistema está funcionando en un cloud y no en local.

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem “Evaluaciones” y la instancia de evaluación correspondiente, que figura bajo el título “Inscripto”.
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en “Agregar”. El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en “Crear equipo”.

Cualquier integrante podrá:

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en “Subir” para iniciar la entrega. Cada equipo (hasta 2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb**

Quando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador de cursos o Coordinación adjunta antes de las 20:00h del día de la entrega, a través de los mails crosa@ort.edu.uy / posada_l@ort.edu.uy (matutino) / larrosa@ort.edu.uy (nocturno), o vía Ms Teams.