

Proyecto de Software 2020 - Trabajo Integrador (TI)

Actividad 4. Aplicación Privada

Fecha de entrega: 18/10 a las 23:59

Todos los años se trata de coordinar el trabajo integrador (TI) de la materia con necesidades y/o demandas de la comunidad. Este año 2020, dada la situación de público conocimiento, nos hemos enfocado en un desarrollo que pueda ser de utilidad tanto a organizaciones como personas individuales que quieran colaborar con la situación provocada por la pandemia debido al COVID-19.

El TI de la cursada se enfocará en el desarrollo de un prototipo que brinde información a los y las ciudadanos/as de la provincia de Buenos Aires, respecto de los centros de ayuda (donación de ropa, de comida, de plasma o de sangre) georeferenciados con el horario de apertura, fases en las que se encuentra cada municipio, etc.

A lo largo de la cursada se desarrollará un prototipo funcional, que es un modelo a escala de la realidad (sin la funcionalidad total), el cual se irá enriqueciendo en cada entrega y que deberán cumplir con los requerimientos y plazos establecidos en el TI.

El trabajo se llevará a cabo en cuatro (4) etapas, en las cuales se deberá entregar un prototipo en funcionamiento en el servidor provisto por la cátedra.

El sistema estará estructurado en dos aplicaciones, aplicación privada y aplicación pública. En la aplicación privada se podrán realizar todas las operaciones administrativas del sistema (CRUD¹ de usuarios, de centros de ayuda social, módulo de configuración del sistema, etc). En la aplicación pública, se visualizará la información de interés para los y las ciudadanos/as, como por ejemplo, dónde se encuentra ubicado el centro de ayuda más cercana, a dónde puede llevar ropa para donar o retirar una bolsa de comida, etc.

¹Por sus siglas en inglés de "Create - Read - Update - Delete": creación, lectura (listado), actualización y borrado de usuarios (ABM).

Proyecto de Software 2020 - Trabajo Integrador (TI)

1.1 Manejo de sesiones

Deberá implementarse un manejo de sesiones adecuado, verificando la sesión y permisos cuando corresponda. Para cada módulo se indicará si requiere autenticación o no y los permisos necesarios.

1.2 Layout

Se deberá implementar el layout de la aplicación que es la base para todas las vistas de la aplicación privada. El resto de las vistas estarán contenidas en este layout sobreescribiendo el contenido central.

La aplicación deberá contar con un menú de navegación que tenga los enlaces a todos los módulos del sistema y esté visible en aquellas vistas del sistema que se consideren necesarias.

Se debe incluir un espacio donde se pueda visualizar la cuenta del usuario con acceso al perfil y link para cerrar la sesión, similar a la Figura 1.

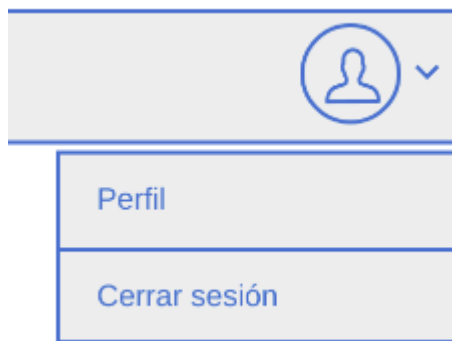


Figura 1. Posible visualización de la cuenta del usuario en el sistema.

También pueden crear un logo para la aplicación y mostrar el mismo en forma coherente en todas las secciones de la aplicación.

1.3 Módulo de usuarios

Desarrollar el **módulo usuarios** que deberá contemplar **al menos** la siguiente funcionalidad:

- CRUD de usuarios: en esta sección deben validar que no existan dos usuarios con el mismo nombre de usuario, la misma validación deberán realizar para el email, es decir, tanto el nombre de usuario como el email son únicos.
- Se considerarán al menos los siguientes datos para cada usuario: nombre, apellido, email, nombre de usuario, activo, password y perfil.
- Se deben poder realizar búsquedas sobre los usuarios, **al menos** por los siguientes campos:
 - nombre de usuario.

Proyecto de Software 2020 - Trabajo Integrador (TI)

- activo/bloqueado.

El resultado de la búsqueda debe estar paginado en base a la configuración del sistema (ver **módulo de configuración**). La validación deberá realizarse del lado del servidor, es decir, la cantidad de registros retornada debe ser la indicada en el módulo de configuración, por ej. 25 registros por página.

- Activar/Bloquear usuario: un usuario bloqueado no podrá acceder al sistema. Se deberá validar que los únicos usuarios que no puedan ser bloqueados, sean aquellos con el rol **Administrador**.
- Asignar o desasignar roles de un usuario, pueden ser varios. En principio se proponen los siguientes roles: administrador del sistema y operador del centro de ayuda.

Para esta etapa, *no será obligatorio desarrollar el CRUD de los roles y los permisos, podrán administrarse desde la base de datos*. Los usuarios, roles y permisos sólo podrán ser administrados por un usuario con rol de **Administrador del sistema**.

Considerar que un usuario podrá tener más de un rol, y para cada rol se pueden configurar varios permisos. Los permisos necesarios asociados a cada rol deberán deducirse del enunciado, ante la duda pueden **consultar a su ayudante**.

El nombre de los permisos deberá respetar el patrón **modulo_accion**, por ejemplo, el módulo de centro de ayuda social (ver **módulo de centro de ayuda social**, siguiente etapa) deberá contemplar los siguientes permisos:

- centro_index: permite acceder al index (listado) del módulo.
- centro_new: permite cargar un centro de ayuda social.
- centro_destroy: permite borrar un centro de ayuda social.
- centro_update: permite actualizar un centro de ayuda social.
- centro_show: permite visualizar un centro de ayuda social.

Es importante entender el porqué del uso de esta solución para implementar la autorización y seguir este esquema de la forma correcta. Esto se explicará oportunamente en los horarios de práctica.

La Figura 2 muestra un posible modelo de usuarios, roles y permisos, que pueden utilizar en el trabajo.

Proyecto de Software 2020 - Trabajo Integrador (TI)

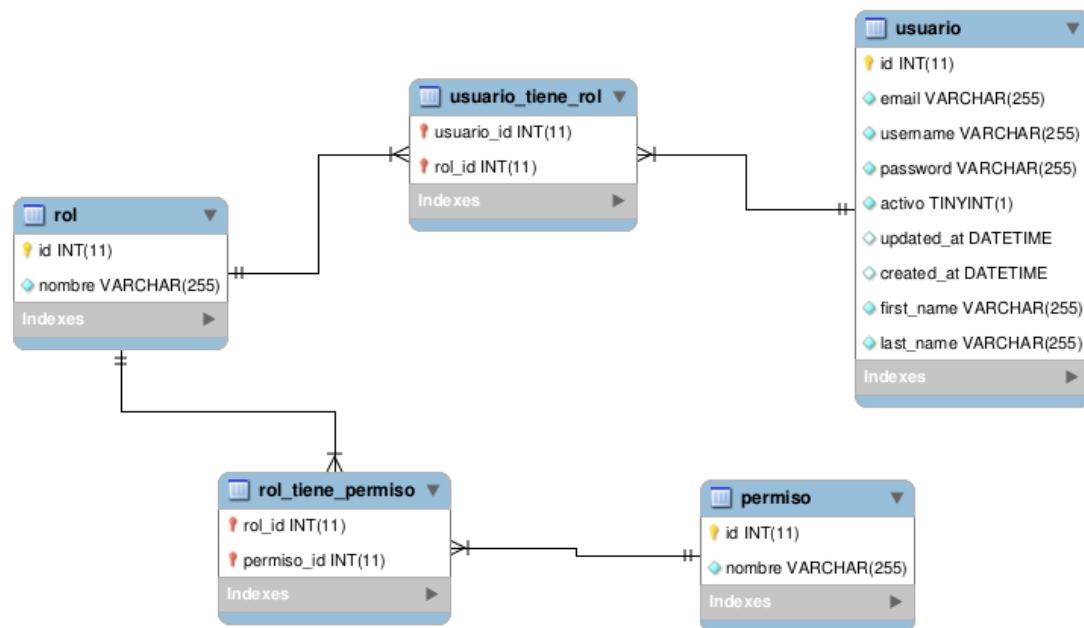


Figura 1. Posible esquema para el manejo de usuarios

1.4 Módulo de configuración

Este módulo permitirá administrar la configuración del sistema, como mínimo deberá contemplar la siguiente configuración:

- Información de página principal
 - Título
 - Descripción
 - Mail de contacto.
- Cantidad de elementos por página en los listados del sistema (todos los listados deberán respetar este valor para el paginado).
- Sitio habilitado/deshabilitado: si se selecciona esta opción, se debe deshabilitar la aplicación pública, mostrando un mensaje configurable como por ejemplo “El sitio se encuentra mantenimiento”. Sólo debe aparecer un enlace hacia el **login** del sistema que deje iniciar sesión a los usuarios con rol **Administrador**.

La configuración del sistema sólo podrá modificarla un usuario con el rol de **Administrador**. Pueden agregar, en caso de que lo consideren necesario y con acuerdo previo con el ayudante asignado, alguna configuración más para ser administrada por este módulo.

Consideraciones generales:

- El prototipo debe ser desarrollado utilizando Python, JavaScript, HTML5, CSS3 y MySQL, y **respetando el modelo en capas MVC**.
- El código deberá escribirse siguiendo las [guías de estilo de Python](#)
- El código Python deberá ser documentado utilizando [docstrings](#).

Proyecto de Software 2020 - Trabajo Integrador (TI)

- El uso de [jinja](#) como motor de plantillas es obligatorio para la aplicación privada.
 - Se debe utilizar [Flask](#) como framework de desarrollo web para la aplicación privada.
 - Se deberán realizar **validaciones de los datos de entrada** tanto del lado del cliente como del lado del servidor. Para *las validaciones del lado del servidor se deben realizar en un módulo aparte* que reciba los datos de entrada y devuelva el resultado de las validaciones. En caso de fallar el controlador debe retornar la respuesta indicando el error de validación.
 - Podrán utilizar librerías que facilitan algunas de las tareas que deben realizar en el trabajo como pueden ser: conexión a servicios externos, librerías de parseo, librerías con patrones para buenas prácticas, validaciones de datos, etc. **Pero todos los miembros del equipo deben demostrar en la defensa pleno conocimiento del funcionamiento de estas librerías y una idea de cómo solucionan el problema.**
 - Para la interacción con la base de datos pueden utilizar un ORM que nos permita además tener una capa de abstracción con la BD. Para esta actividad no será obligatoria la utilización de un ORM, pero sí lo será en las Actividades 5, 6 y 7.
 - No pueden utilizar un framework/generador de código para el desarrollo de la API de Instituciones.
 - Debe tener en cuenta los conceptos de Semántica Web proporcionada por HTML5 siempre y cuando sea posible con una correcta utilización de las etiquetas del lenguaje.
 - El trabajo será evaluado desde el servidor de la cátedra que cada grupo deberá gestionar mediante Git. **NO se aceptarán entregas que no estén realizadas en tiempo y forma en el servidor provisto por la cátedra.**
 - Deberán visualizarse los aportes de cada uno/a de los/as integrantes del grupo de trabajo.
 - El/la ayudante a cargo evaluará el progreso y la participación de cada integrante mediante las consultas online y el seguimiento mediante GitLab.
 - Toda vista (**HTML5** y **CSS3**) debe validar contra las especificaciones de la W3C (<http://validator.w3.org/> y <https://jigsaw.w3.org/css-validator/> respectivamente). En esta oportunidad puede utilizar **Bootstrap** u otro framework similar. En caso de que alguna de las vistas no valide, deberá realizar un breve informe indicando cuales son los errores encontrados.
 - Todas las vistas deben cumplir ser web responsive y visualizarse de forma correcta en distintos dispositivos. Al menos deben contemplar 3 resoluciones distintas:
 - < 360
 - 360 < res < 768
 - > 768
- Esta pauta no es obligatoria para esta actividad, pero lo será para las siguientes.
- La entrega es obligatoria. Todos y todas los/as integrantes deben presentarse a la defensa.
 - El sistema no debe ser susceptible a SQL Injection, XSS ni CSRF.
 - **Importante:**
 - El proyecto podrá ser realizado de modo individual o en grupos de tres o cuatro integrantes (será responsabilidad de los y las estudiantes la conformación de los equipos de trabajo). Todos y todas los/as estudiantes cumplirán con la totalidad de la consigna, sin excepciones.

Proyecto de Software 2020 - Trabajo Integrador (TI)

Información del servidor

Tener en cuenta que el servidor de la cátedra utiliza los siguientes servicios y versiones:

- Servidor de Base de Datos: MariaDB 10.1.44
- Intérprete Python: Python 3.6.9
- Servidor web: Apache 2.4.29 (Ubuntu)