

Universidad de San Carlos de Guatemala Facultad de  
Ingeniería  
Escuela de Ciencias y Sistemas Análisis y  
Diseño de Sistemas 1  
Ing. Allan Alberto Morataya Ramos  
Aux. Cesar Sazo  
Sección: “A+”



## Practica No. 2 Pruebas Unitarias

### Contenido

Objetivos.	2
Generales	2
Específicos	2
Descripción	2
Login	3
Registro	3
Página Principal	4
Agregar Amigos	4
Datos Personales	5
Testing	5
Consideraciones	6
Entregables.	6
Fecha de Entrega	6

## Objetivos

### Generales

- Conocer y entender el concepto de pruebas unitarias.
- Comprender la importancia del trabajo en equipo

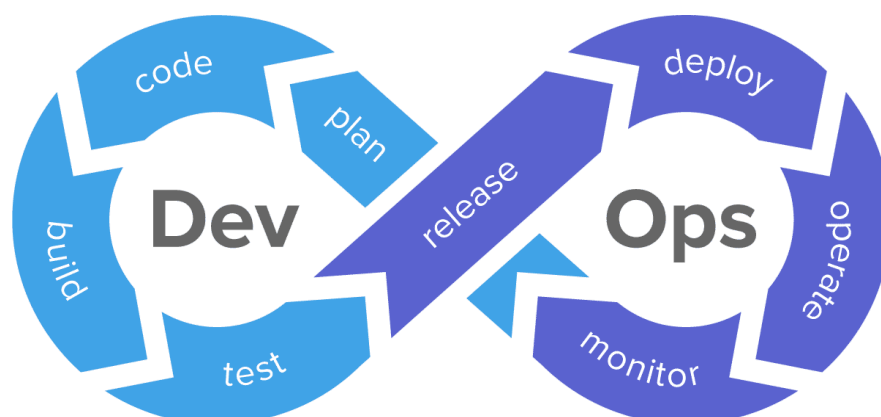
### Específicos

- El estudiante podrá listar las diferentes pruebas que se pueden realizar en un proyecto de software.
- Utilizar una herramienta para realizar pruebas unitarias
- Aplicar al menos 5 pruebas unitarias por funcionalidad.

## Descripción

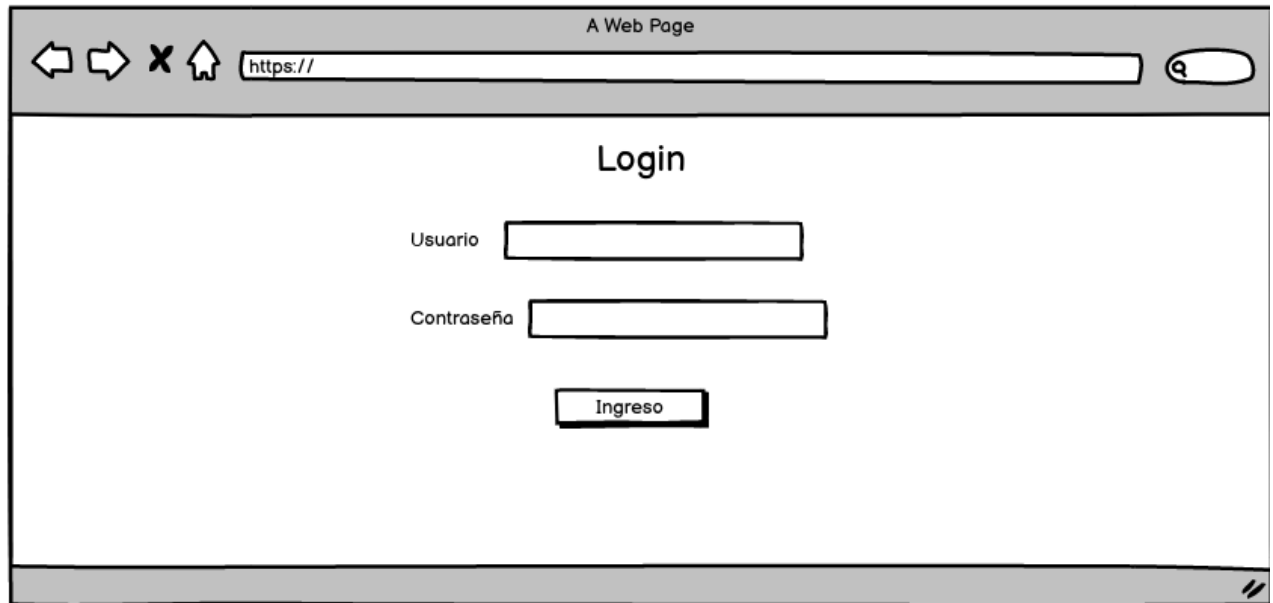
La práctica consiste en la creación de una aplicación web, tipo red social. La cual tendrá funciones básicas, como inicio de sesión, postear fotos y agregar amigos (estas se detallan más adelante). La parte importante es la aplicación de un marco de trabajo y la filosofía de una constante evolución con el fin de promover un mejor desarrollo de aplicaciones en menos tiempo y la rápida publicación de nuevas o revisadas funciones de software o productos para los clientes, todo esto se conoce como **DevOps**, asimismo aplicando la metodología de Scrum con el fin de acelerar y mejorar el desarrollo y las publicaciones de productos, el uso de pruebas unitarias para cada una de las funcionalidades antes mencionadas.

A continuación, se describen las secciones que contendrá la aplicación web:



## Login

Para el loguearse, se debe de ingresar su nombre de usuario y contraseña y una confirmación de contraseña para validar que sean correctos



A Web Page

https://

### Login

Usuario

Contraseña

## Registro

Para registrar un nuevo usuario se deberán ingresar los siguientes datos.

- Usuario
- Nombre Completo
- Contraseña
- Confirmar Contraseña
- Cargar Imagen de foto de perfil



A Web Page

https://

### Registro

Usuario

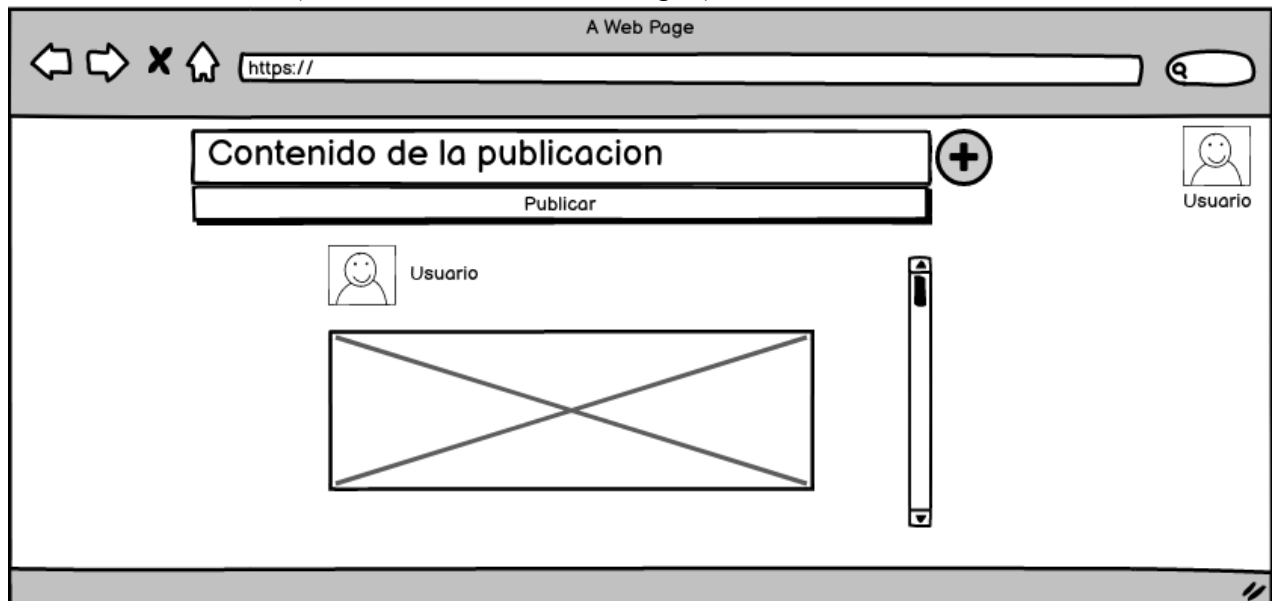
Contraseña

## Página Principal

### Publicaciones

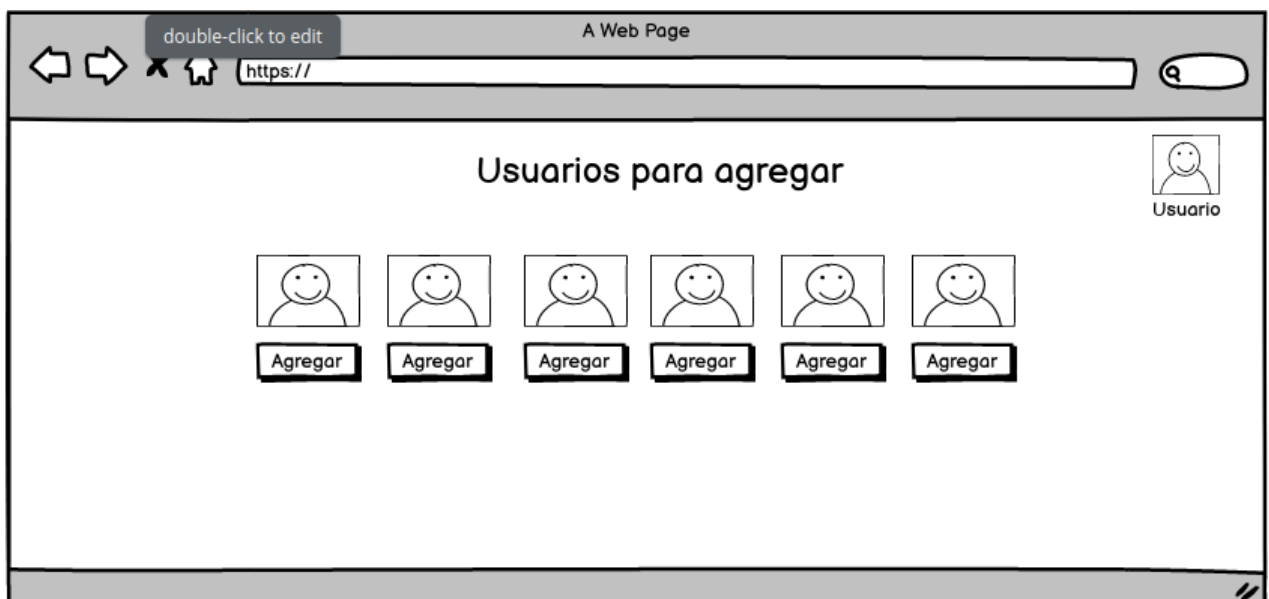
En este apartado el usuario podrá crear una publicación, la cual puede contener imagen, texto o ambas.

Así mismo se muestran todas las publicaciones que otros usuarios han realizado que tienen amistad con dicho usuario con inicio de sesión, al igual que también podrá ver lo que el mismo usuario ha publicado. Las publicaciones se deben de mostrar por orden de fecha de publicación de forma descendente (del más reciente al mas antiguo).



### Agregar Amigos

En este apartado el usuario podrá agregar como amigos a otros usuarios en la aplicación, en la sección de amigos agregados tendrán que aparecer solo los amigos que han aceptado la solicitud de amistad enviada.



## Datos del Perfil

El usuario puede ver y modificar sus datos, en este caso será:

- Nombre completo
- Nombre de usuario
- Foto de perfil
- Contraseña

Tiene que ingresar la contraseña correcta para que se guarden los datos modificados

## DEVOPS

Implementación de la cadena de herramientas de DevOps con el fin de optimizar, acortar y automatizar las diversas etapas del flujo de trabajo de creación de software, para dicha practica es importante aplicar estas herramientas que se emplean en las diversas etapas del ciclo de DevOps.

- **Planificación (SCRUM)** En esta fase se definen los requisitos y valores de la práctica. Algunas herramientas de muestra o tableros son Jira o Git, con las cuales se puede hacer un seguimiento de los problemas conocidos y llevar a cabo la gestión de los módulos de la práctica.
- **Codificación.** Esta fase implica el diseño del software y la creación del código. Algunas herramientas de muestra son GitHub, GitLab, Bitbucket o Stash.
- **Compilación.** En esta fase se gestionan las versiones y las compilaciones del software, y se utilizan herramientas automatizadas que ayudan a compilar y crear paquetes de código para publicarlos después para la producción. Se utilizan repositorios de código fuente o repositorios de paquetes que también proporcionan la infraestructura que se necesita para el lanzamiento del producto. Algunas herramientas de muestra son Docker, Ansible, Puppet, Chef, Gradle, Maven o JFrog Artifactory.
- **Prueba. (TESTING)** Esta fase incluye la realización de pruebas para garantizar la calidad de la programación. Algunas herramientas de muestra son JUnit, Codeception, Selenium, Vagrant, mocha y chai, TestNG o BlazeMeter.
- **Puesta en marcha.** En esta fase se emplean herramientas que ayudan a gestionar, coordinar, programar y automatizar las tareas de producción de las versiones de productos. Algunas herramientas de muestra son Puppet, Chef, Ansible, Jenkins, Kubernetes, OpenShift, OpenStack, Docker o Jira.
- **Funcionamiento.** En esta fase se gestiona el software durante su producción.
- **Supervisión.** En esta fase se identifica y recopila información sobre problemas que surgen en una versión de software específica que se encuentra en producción.

## TESTING

Para la parte de testing, se deben de crear distintas pruebas unitarias para verificar el correcto funcionamiento de la aplicación.

El número de pruebas realizadas debe ser igual al número de métodos que se implementen. Por ejemplo si se crean 20 métodos, deben de escribirse 20 pruebas unitarias.

Deben utilizar un software o herramienta que les mida el code coverage.

El code coverage debe ser aplicado a todo el código, sin ignorar líneas.

## SCRUM (metodología de DevOps)

Se debe utilizar la metodología SCRUM para el desarrollo de la aplicación.

- Realizar una grabación con todos los integrantes del equipo haciendo los tipos de reuniones (Sprint Planning Meeting, Daily Scrum meeting, Sprint Review meeting y Retrospective Meeting).
- Cualquier herramienta para la organización de la práctica (trello, jira etc.) agregar al auxiliar: checha18964@gmail.com
- Enviar las grabaciones con 2 días de anticipación de la entrega de la práctica 2.

## Consideraciones que deben tomar en cuenta

Se deberá trabajar con los grupos formados en el laboratorio.

- Se debe trabajar haciendo uso de gitflow
- Agregar el siguiente usuario de como colaborador para revisión de commit trabajado github(@cesar-sazo) gitlab: (@cesar-sazo)
- El lenguaje a utilizar queda a discreción del grupo de trabajo.
- Cada “commit” debe tener un mensaje describiendo la acción, al final de
- Cada mensaje debe ir incluido en el carné.
- Dudas y preguntas serán respondidas en el correo electrónico o foros de UEDI.
- Las copias totales y parciales tendrán una nota de 0 y será reportada a la escuela de sistemas.
- El manual técnico debe incluir en el Readme del repositorio.
  - Casos de Pruebas documentados
  - Descripción de pruebas implementadas

Entregables:

- Documento PDF con nombre: [AyD1]Práctica 2\_#grupo.pdf (Integrantes y Link del repositorio)
- Entregar vía UEDI (una persona por grupo).

Fecha de entrega: 28-03-2022; a las 23:59.

Fecha y hora de calificación: Día y hora pendientes.

Lugar: Google Meet.