

Universidad ORT Uruguay

Facultad de Ingeniería

Escuela de Tecnología

## OBLIGATORIO PROGRAMACIÓN 1

### Informe de testing



Cristian Poggi - 203814



Emanuel Díaz - 275164



Marcos Medina - 251153

# Test Plan GuitarApp

Dejaremos aquí evidencia de todos los casos de prueba utilizados para garantizar la calidad de nuestra aplicación. Al principio dejo los test de regresión basados en el documento de análisis para luego dar lugar a los bugs encontrados en las últimas validaciones de calidad para guitarApp.

## Registro de usuarios

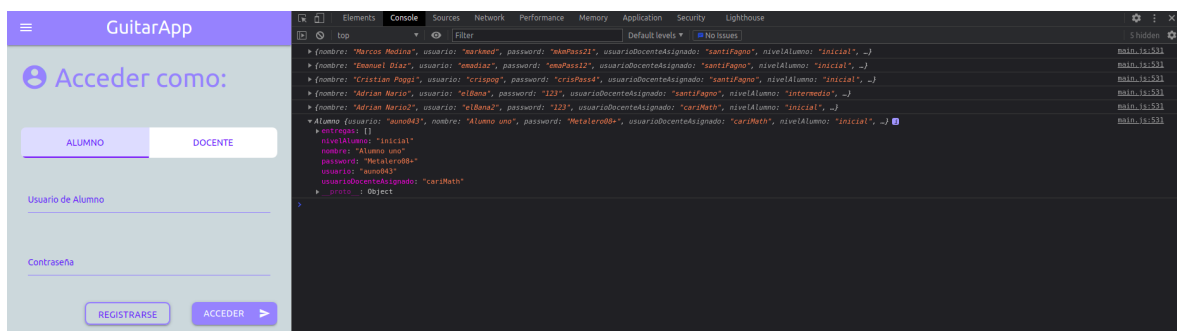
Registro de usuario Alumno.

Caso de prueba creado para validar el registro de los alumnos. Actualmente contamos con una función dentro de la función de registro de alumnos la cual se encarga de listar todos los usuarios que tenemos registrados hasta el momento. Incluyendo el último que registramos al momento de hacer la prueba.

Precondiciones:

- Tener docentes registrados para poder seleccionar 1
- Hay validaciones a cumplir.
  - No se permiten campos vacíos en nombre y contraseña.
  - El nombre no puede tener símbolos.
  - La contraseña tiene que tener al menos 4 caracteres, al menos una mayúscula, una minúscula y un número.

Resultados esperados - El usuario es registrado en el array alumnosList y es mostrado en la consola junto con los anteriores usuarios registrados.



Pasos

1. Vamos al formulario de registro
2. Seleccionamos la pestaña de Alumnos

- |  |
|--|
| <ol style="list-style-type: none"><li>3. Ingresamos los datos solicitados siguiendo las precondiciones</li><li>4. Damos click en registrar</li></ol> |
|--|

Caso de prueba - Registro de usuario Docente
--

Precondiciones - Abrir el sistema y tener en cuenta las siguientes validaciones
---

- |   |
|---|
| <ul style="list-style-type: none"><li>- Validaciones a cumplir para ingresar Nombre, Usuario y Contraseña.<ul style="list-style-type: none"><li>• No se permiten campos vacíos en nombre y contraseña.</li><li>• El nombre no puede tener símbolos.</li><li>• La contraseña tiene que tener al menos 4 caracteres, al menos una mayúscula, una minúscula y un número.</li></ul></li></ul> |
|---|

Resultados esperados - El usuario es registrado en el array docentesList
--

Pasos
-------

- |   |
|---|
| <ol style="list-style-type: none"><li>1. Vamos al formulario de registro</li><li>2. Seleccionamos la pestaña de Docentes</li><li>3. Ingresamos los datos solicitados siguiendo las precondiciones</li><li>4. Damos click en registrar</li></ol> |
|---|

## Login/Logout

Caso de prueba - Login de usuario Alumno
--

Precondiciones - Contar con un usuario Alumno registrado
--

Resultados esperados - El usuario ingresa a la App con éxito quedando ubicado en el dashboard de docentes.
--

Pasos
-------

- |   |
|---|
| <ol style="list-style-type: none"><li>1. Ir a la pestaña de login de alumno</li><li>2. Ingresar Nombre</li><li>3. Ingresar Contraseña</li><li>4. Dar click en login</li></ol> |
|---|

## Caso de prueba - Login de usuario Docente

Precondiciones - Contar con un usuario Docente registrado

Resultados esperados - El usuario ingresa a la App con éxito quedando ubicado en el dashboard de alumnos.



## Pasos

1. Ir a la pestaña de login de Docente
2. Ingresar Nombre
3. Ingresar Contraseña
4. Dar click en login

Caso de prueba - Logout del usuario Alumno
Precondiciones - Contar con un usuario Alumno logueado
Resultados esperados - El usuario se desloguea de la App con éxito y queda parado en el menú de acceso.
Pasos <ol style="list-style-type: none"> <li>1. Dentro del dashboard del alumno damos click en Cerrar sesión.</li> </ol>

Caso de prueba - Logout del usuario Docente
Precondiciones - Contar con un usuario Docente logueado
Resultados esperados - El usuario se desloguea de la App con éxito y queda parado en el menú de acceso.
Pasos <ol style="list-style-type: none"> <li>1. Dentro del dashboard del alumno damos click en Cerrar sesión.</li> </ol>

## Funcionalidades de los Dashboard para ambos usuarios

Caso de prueba - Ver ejercicios planteados (Alumno)
Precondiciones - Estar logueado como usuario alumno y contar con datos para realizar la prueba
Resultados esperados - Veremos los ejercicios en el dashboard en la parte de tareas
Pasos <ol style="list-style-type: none"> <li>1. En el dashboard veremos el total de los ejercicios planteados en la parte inferior en el sector de tareas.</li> </ol>

Caso de prueba - Realizar entrega de un ejercicio (Alumno)
--

Precondiciones - Estar logueado como alumno y contar con una tarea planteada por el profesor.

Resultados esperados - Se cargan los datos del alumno a la tarea y queda pronta para que el Docente la corrija.

#### Pasos

1. Dentro del dashboard de alumno abrimos la tarea
2. LLenamos los campos pertinentes para realizar la entrega
3. Click en entregar tarea

#### Caso de prueba - Ver ejercicios resueltos (Alumno)

Precondiciones - Estar logueado como usuario Alumno y contar con tareas planteadas para ver la estadística en el dashboard

Resultados esperados - El usuario tendrá esta estadística en el dashboard como tareas totales.

Caminos posibles para reproducción de la prueba

Si no estamos logueados, loguearnos como Alumno.

Si estamos logueados veremos el dato en el dashboard del alumno.

#### Caso de prueba - Asignar un nivel al alumno (Docente)

Precondiciones - Estar logueado como usuario Docente. Desde el reporte individual del alumno realizaremos la modificación.

Resultados esperados - Asignar un nivel a un alumno teniendo en cuenta la única validación, no se puede asignar nivel previo.

#### Pasos

1. Dentro del reporte individual
2. Elegimos alumno al cual le asignaremos el nivel
3. Seleccionamos nivel

4. Click en botón cambiar

Caso de prueba - Plantear ejercicios (Crear tarea)

Precondiciones - Estar logueado como profesor

Resultados esperados - Un registro completo de los datos en el array de tareas. Los datos deben cumplir las validaciones pertinentes y se mostraría un mensaje en pantalla para asegurarle el registro de sus datos al docente.

Título

Titulo de la tarea345

Descripción

Descripcion de la tarea2354

Nivel

Intermedio

Imagen

Imagen Ejercicio3

CREAR

← VOLVER

CREAR TAREA

✓ Se ha registrado una nueva tarea

Pasos

1. En el dashboard del docente damos click al boton Tareas
2. Click en Botón crear
3. Llenamos el formulario con los datos pertinentes (Título, descripción Nivel e imagen)
4. click en botón crear

Caso de prueba - Proceso de devoluciones
Precondiciones - Estar logueado como usuario Docente y contar con tareas entregadas por el Alumno para poder realizar la devolución
Resultados esperados - Se realiza la corrección pertinente dejando todos los datos para que el Alumno las consuma más tarde
Pasos <ol style="list-style-type: none"> <li>1. Dentro del dashboard vamos a la pestaña de pendientes.</li> <li>2. Damos click en Entregar a la tarea planteada que queremos entregar.</li> <li>3. Ingresamos los datos en el form.</li> <li>4. Damos click en Entregar Tarea</li> </ol>

## Estadísticas Alumno

Caso de prueba - Reporte de ejercicios resueltos (Porcentaje) F06
Precondiciones - Contar con datos hardcodeados en db.js o registrar nuevos. La primera versión contaría con data hardcodeada. Recordar estar logueado como usuario alumno.
Resultados esperados - Veremos la estadística en el dashboard. El dato se mostrará como porcentaje (%)
Caminos posibles para reproducción de la prueba  Si no estamos logueados, loguearnos como Alumno.  Si estamos logueados veremos el dato en el dashboard del alumno.

Caso de prueba - Reporte de devoluciones (Tareas corregidas)
Precondiciones - Estar logueado como alumno
Resultados esperados - Ver en el dashboard del Alumno las tareas corregidas
Caminos para reproducir la prueba de este reporte.  Si no estamos logueados, debemos loguearnos como ususario Alumno



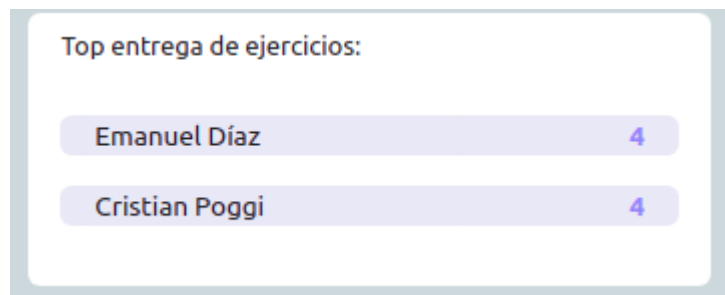
Si estamos logueados como usuario alumno veremos el reporte en el dashboard, tareas realizadas.

## Estadísticas Docente

Caso de prueba - Reporte de alumno con más ejercicios resueltos F13.1

Precondiciones - Estar logueado como Docente y contar con datos para mostrar en las estadísticas.

Resultados esperados - Ver en el dashboard una lista de los Alumnos con más entregas



Pasos para la reproducción serían:

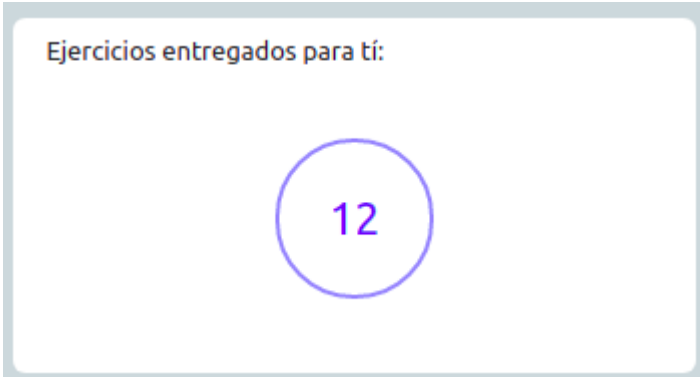
Si no estamos logueados nos logueamos como docente.

En caso que estemos logueados, deberíamos ver esta estadística reportada dentro de una lista en el dashboard del Docente en la parte inferior izquierda.

Caso de prueba - Reporte de cantidad de ejercicios entregados (Docente) F13.2

Precondiciones - Estar logueado como Docente y contar con datos para mostrar en las estadísticas.

Resultados esperados - Vemos en el dashboard la estadística pertinente a los ejercicios entregados por parte de los alumnos


<p><b>Pasos</b></p> <p>Si no estamos logueados nos logueamos como docente.</p> <p>En caso que estemos logueados, deberíamos ver esta estadística reportada dentro de un chart en el dashboard.</p>

Caso de prueba - Reporte individual de alumno F13.3
Precondiciones - Estar logueados como usuario docente desde el dashboard.
Resultados esperados - El usuario lograría ver los datos para un alumno individualmente
<p><b>Pasos</b></p> <ol style="list-style-type: none"> <li>1. Parados en el dashboard damos click en [Reporte individual]</li> <li>2. Seleccionamos al alumno que deseamos ver sus estadísticas individuales.</li> </ol>

## Casos a considerar en regresión del test plan.

Dejaremos evidencia de algunos casos borde que notamos al realizar el testing que generan alguna inconsistencia en la UI. Incluiremos revisiones, ajustes, imágenes, evidencias.

Caso de prueba - Login con campos vacíos para ambos usuarios
Precondiciones - Estar logueados como usuario docente/alumno desde el form del login.

Resultados esperados - Dadas las validaciones debemos ver errores en forma de toasts de materialize los cuales nos imprimirán en pantalla mediante una función los errores pertinentes a cada validación sea correcta o no.

Actualmente contamos con un escenario donde se nos solapan los Toast si van varias validaciones sin cumplir.

**Resgistrarse como:**

**ALUMNO**    **DOCENTE**

Nombre de Alumno  
Alumno uno

Usuario de Alumno  
auno043

Contraseña

Seleccione Docente  
Choose your option

**REGISTRARSE**

Contraseña inválida. Corrija: cantidad de caracteres, al menos una minúscula, al m

Por favor, seleccione un Docente

Pasos para reproducción, recordar que hay dos tipos de login uno para el alumno y otro para el docente. Utilizar este TC para probar ambos logueos al sistema.

Caso de prueba - Verificar limpieza de los formularios de registro para ambos usuarios

Precondiciones - Estar parados en el sobre el formulario de registro y haber registrado un usuario previamente

Resultados esperados - El formulario debería estar vacío luego de haber registrado un usuario.

Pasos para la reproducción serían estar parados en el formulario de registro y ahí llenar los datos pertinentes para registrar un Docente o un Alumno. Al registrar el usuario deberíamos ver el formulario con los campos vacíos para dejar que el usuario los complete nuevamente si es que quiere registrar alguno mas.

Caso de prueba - Limpieza de los formularios de login para ambos usuarios

Precondiciones - Estar parados en el sobre el formulario de login

Resultados esperados - Al cerrar la sesión verificar si el formulario de login cuenta con los campos vacíos.

Para reproducirlo la idea sería hacer login con un usuario, cerrar sesión y ver si el formulario de login está vacíos

Caso de prueba - Limpieza de los formularios de Entrega de Tareas

Precondiciones - La idea es estar logueados en el sistema como docentes y registrar una tarea.

Resultados esperados - Esperamos que este reporte esté vacío cuando el docente quiera ingresar otra tarea.

No está contemplado en versión a entregar en el obligatorio pero notamos este ajuste par mejorar la calidad.

# Crear Tarea

Título

Titulo de la tarea

Descripción

Descripcion de la tarea

Nivel

Inicial

Imagen

Imagen Ejercicio3

CREAR

Caso de prueba - Limpieza de los formularios de Entrega de Tareas

Precondiciones - La idea es estar logueados en el sistema como alumno y ver el dashboard limpio de datos que no correspondan a los reportes

Resultados esperados - Esperamos que el dashboard esté limpio.



Para reproducirlo debemos loguearnos como alumno y ver los detalles de cada reporte. Podríamos a futuro cubrir reporte por reporte la limpieza de los datos

**Solución:** Se crearon las variables `cantidadTareasTotal` tareas totales en el main y `numeroTareasRealizadas` para tareas realizadas.

## LISTA DE BUGS ENCONTRADOS POST-ENTREGA

1. Bug del botón volver extra en dashboardAlumno (el de Materialize no funciona, el común HTML Sí)

**Solución:** Borrar el botón extra que se nos olvidó borrar.

2. Bug de que te sale un error en vez de un Success cuando ya entregaste alguna vez antes alguna tarea (Alumno). Pero se pushea con éxito. (**SOLO SALE EL TOAST QUE NO VA, EL DE ERROR EN VEZ DE SUCCESS**). El bug es originado al momento de ingresar a la sección. Al entrar se le asigna la función al botón pero sin remover cualquier funcionalidad anterior. Esto provoca que se ejecute una misma función la cantidad de veces que se ingrese a la sección.

**Solución:** Hacer un `removeEventListener` al botón al momento de ingresar a la sección, para así remover cualquier funcionalidad anterior.

3. Usuario y contraseña no se limpian después de cerrar sesión (Es relevante?).

3. B. Tampoco se limpia el formulario del ejercicio/devolución.

**Solución:** Implementar un document querySelector que limpie el .value con = ""

4. Al pasar de nivel a un alumno; su número de entregasRealizadas en su dashboard sigue intacto, lo demás se actualiza perfectamente (**esto es porque al cambiar de nivel a un alumno no se limpian las tareasRealizadas que tenía en su nivel anterior**). Las tareas realizadas del nivel anterior ya no se muestran xq son de otro nivel, pero sus tareasRealizadas en Stats si las tienen en cuenta.

**Solución:** Poder mantener las tareas ya asignadas previamente en el DASHBOARD y actualizar (o refrescar) las nuevas, para tener todas en el dashboard. *Que siga mostrando los objetos de las tareas realizadas que realizó el alumno antes de subir de Nivel.*

5. Sigue habiendo un boton de verPerfil al lado de los alumnos en DashBoardDocente, que entre nosotros sabíamos que no pasaba nada, pero el profe lo puede criticar perfectamente. Ese botón no debería estar ahí porque genera un bug innecesario cuando en realidad la función si anda bien que es el reporte individual cuyando clickeas en reporte individual(no Tan relevante ya que salta un mensaje de alerta diciendo WIP; vaya a reporte Individual)

**Solución:** Borrar el botón ver perfil que sobra al lado de cada item de alumno en la lista del dashboard. O asignarle la funcionalidad al mismo (funcionalidad que YA está pero en otro lado) de mostrar los datos del alumno específico.

6. **Un bug cuando se hacen devoluciones:** NO PUSHEA LA DEVOLUCIÓN. LA FUNCIÓN ESTÁ HECHA PERO **LÍNEA 601 & LÍNEA 606** (Main.js) CUANDO DETECTA SI EXISTE UNA DEVOLUCIÓN, NO FUNCIONA PORQUE HAY UN ERROR DE TIPO.

```
598 comentario, comentario = LLLL
599 )
600 for (const Entrega of alumno.entregas) { alumno = {nombre: "Emanuel Díaz", usuario: "ema
601   if (entrega.id_tarea == idTareaCorregir) { idTareaCorregir = "1"
602     entrega.devolucion = nuevaDevolucion
603   }
604 }
605 for (const DtareaDocente of DgetDocente(alumno.usuarioDocenteAsignado).tareasPlanteadas) {
606   if (tareaDocente.idTarea == idTareaCorregir) {
607     for (const Entrega of tareaDocente.entregas) {
608       if (entrega.usuarioAlumno == usuarioAlumno) { // SI MATCHEA QUIERE DECIR QUE ESTOY COR
609         entrega.devolucion = nuevaDevolucion
610       }
611     }
612   }
}
```

ESTAMOS COMPARANDO UN STRING CON UN NUMBER, ERROR DE TIPO

**Solución:** Pasar a **Number**. Este problema nos pasó varias veces pero en esta función se nos olvidó corregirlo.

7. **(NO-BUG) Faltó la función de SEARCH.**

**Solución:** Lo creamos después de entregar

JS main.js

JS busqueda.js X

JS busqueda.js > buscarTarea

```
1 // Subrutina de buscar elementos
2 function buscarElemento(arrElementos, propiedad, busqueda) {
3     let existe = false;
4     for (let i = 0; i < arrElementos.length; i++) {
5         const unElemento = arrElementos[i];
6         if (unElemento[propiedad] === busqueda) {
7             existe = true;
8             break;
9         }
10    }
11    return existe;
12 }
13 // Funcion de buscar la tarea, según el tipo de Tarea que estemos queriendo buscar
14 // SOLO FUNCIONARIA PARA ENCONTRAR ABSOLUTAMENTE TODAS LAS TAREAS ASIGNADAS AL ALUMNO
15 function buscarTarea (tipoTareas) {
16     input = document.querySelector("#inputBusqueda")
17     docente = getDocente (datausuarioLogeado.usuarioDocenteAsignado)
18     const tareasAlumno = docenteAsignado.tareasPlanteadas.filter((tarea) => tarea.nivel === dataUsuarioLogeado.nivelAlumno)
19
20     for (tarea of tareasAlumno)
21     if (buscarElemento(tareasAlumno, titulo, input)) {
22         tareasEncontradas.push(tarea)}
23     tipoTareas.innerHTML = tareasEncontradas.push
24
25     if (tareasEncontradas.length < 1) {
26         if (buscarElemento(tareasAlumno, descripcion, input)) {
27             tareasEncontradas.push(tarea)}
28             tipoTareas.innerHTML = tareasEncontradas.push
29     }
30
31     if (tareasEncontradas.length < 1) {
32         | mostrarAlerta("No se encontró tareas", "error")
33     } else if (tareasEncontradas.length > 0) {
34         mostrarAlerta("Se encontraron tareas para su búsqueda", "success")
35
36     }
37 }
38
```