

```

@Override
public void update(Observable o, Object arg) {
    switch ((String) arg) {
        case "Estudiante":
            binding.lavCargando.setVisibility(View.INVISIBLE);
            Toasty.error(context: this, message: "Los estudiantes no tienen acceso al sistema", Toasty.LENGTH_LONG, withIcon());
            break;

        case "Administrador":
            startActivity(new Intent(packageContext: this, NavigationDrawerActivity.class));
            finish();
            break;

        case "Error login":
            binding.lavCargando.setVisibility(View.INVISIBLE);
            binding.etlUsuario.setError("Usuario o contraseña incorrecta");
            binding.etlClave.setError("Usuario o contraseña incorrecta");
            break;

        case "Error servidor":
            binding.lavCargando.setVisibility(View.INVISIBLE);
            binding.etlIp.setError("IP del servidor incorrecta");
            break;
    }
}
}

```

El Proyecto se hizo con los patrones *MVC* y *Observer*, por lo tanto, cuando el *Observable* hace un cambio se notifica al activity y dependiendo de argumento se hará una acción única. En el caso de que el login sea correcto, se envía al usuario al *NavigationDrawerActivity*.

```

android {
    compileSdkVersion 29
    buildToolsVersion "29.0.3"
    viewBinding.enabled = true
}

```

En el proyecto se utilizó *viewBinding* que sirve para no tener que asociar los objetos de una vista en el activity con el *findViewById*, por lo que ahorra mucho tiempo y código repetitivo.

```

private FragmentCursosBinding binding;
private CursosController cursosController;
private int position;
private boolean deshacer = false;

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    binding = FragmentCursosBinding.inflate(inflater);
    setHasOptionsMenu(true);

    binding.fabAgregarCurso.setOnClickListener(v -> startActivityForResult(new Intent(getContext(), CrearCursoActivity.class)));
    binding.srlCursos.setOnRefreshListener(this);

    cursosController = new CursosController(new CursosModel(), cursosFragment: this);
}

```

En esta imagen se puede observar como funciona el viewBinding, el objeto *binding*, de tipo *FragmentCursosBinding*, contiene todos los objetos del xml asociado con este *fragment*.

```
118 binding.rvCursos.setAdapter(cursosController.getCursosModel());
119 binding.rvCursos.setLayoutManager(new LinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL, reverseLayout: false))
120 binding.rvCursos.addItemDecoration(new DividerItemDecoration(getContext(), DividerItemDecoration.VERTICAL));
```

En estas tres líneas es donde se asignar el *Adapter* al *Recycler View*, también se asigna el *LayoutManager*, que será de tipo vertical, por ultimo se le agrega un decorador, que es una línea divisoria entre ítems.