Ingeniería de Software

Trabajo Práctico 2

Arquitectura del sistema

Proyecto “TBreaker”

|  |  |
| --- | --- |
| Profesor: | Julián Nonino |
| Grupo: | Fuplace |
| Integrantes: | Vignolo, Gabriel Enrique |
|  | Losano Quintana, Juan Cruz |
|  | Velazquez, Cristian |

**Historial de revisión**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Sumario de cambios** | **Autor** |
| 1.0.0 | 23/05/18 | Inicio documento de arquitectura del sistema | Losano Quintana, Juan Cruz  Velazquez, Cristian  Vignolo, Gabriel Enrique |
| 1.0.1 | 23/05/18 | Gráfico de arquitectura y Patrón de arquitectura. | Losano Quintana, Juan Cruz  Velazquez, Cristian  Vignolo, Gabriel Enrique |
| 2.0.0 | 26/05/18 | Diagrama de despliegue y diagrama de componentes. | Losano Quintana, Juan Cruz  Velazquez, Cristian  Vignolo, Gabriel Enrique |

**Tabla de Contenidos**

[1 Arquitectura general 4](#_gjdgxs)

[1.1 Gráfico de arquitectura 4](#_30j0zll)

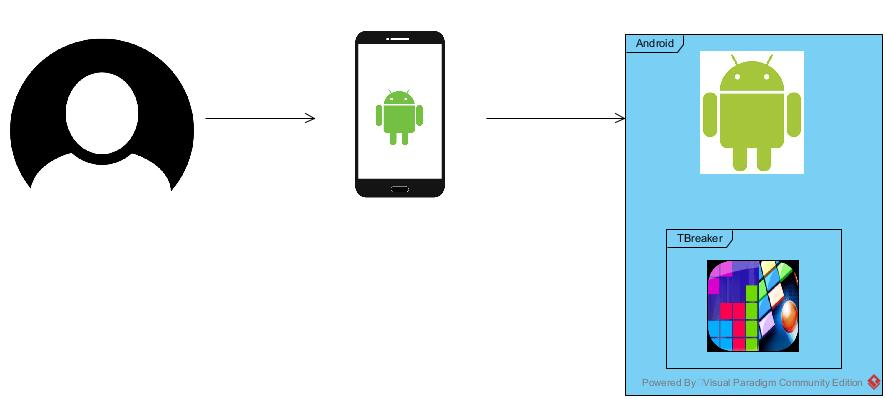
[1.2 Patrón de arquitectura 4](#_1fob9te)

[2 Diagramas](#_3znysh7) 5

[2.1 Diagrama de Despliegue](#_2et92p0) 5

[2.2 Diagrama de Componentes](#_2s8eyo1) 6

# Arquitectura General



## **Gráfico de arquitectura**

## **Patrón de arquitectura**

MVC (Model View Controler): Es un patrón de arquitectura que separa la presentación y la interacción de los datos del sistema. Tiene una estructura compuesta por tres componentes lógicos relacionados entre sí. Los componentes son los siguientes:

*Modelo:* se encarga de manejar los datos del sistema y los métodos y operaciones asociados a esos datos. Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la “vista” aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al “modelo” a través del “controlador”.

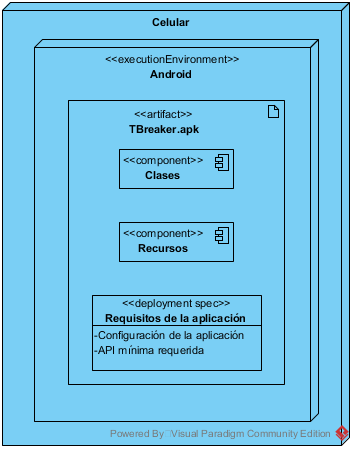
*Vista:* Define la presentación de los datos hacia el usuario. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

*Controlador:* Dirige la interacción del usuario y pasa estas interacciones a vista y modelo. Las interacciones pueden ser por ejemplo: cuando se presiona una tecla. También puede enviar comandos a su “vista” asociada si se solicita un cambio en la forma en que se presenta el “modelo”.

Se va a utilizar MVC ya que permite facilitar el mantenimiento de la aplicación. Por ejemplo: si lo que queremos es modificar la apariencia de la aplicación sólo modificaremos el componente Vista; si queremos cambiar de sistema de almacenamiento de datos sólo tendremos que modificar el componente Modelo; y si queremos modificar el código para optimizar rendimiento lo que haremos será modificar la capa del Controlador.

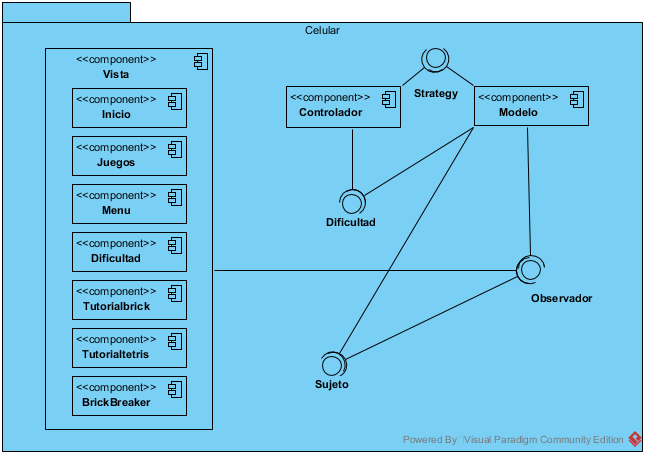
# Diagramas

## **Diagrama de despliegue**



El diagrama de despliegue de UML sirve para modelar la disposición física de los artefactos del software y su relación con los elementos hardware. En el caso presente se tiene un celular, el mismo tiene un entorno de ejecución en Android en donde correrá la aplicación (TBreaker.apk). Esta última tiene distintos componentes como las clases que la integran y las especificaciones de despliegue.

## **Diagrama de componentes**



El diagrama de componentes muestra cómo un sistema de software es dividido en componentes y muestra las dependencias entre los mismos. Esta expone la interacción entre el usuario, la vista, el controlador y el modelo. El primero provee entradas a las distintas pantallas dispuestas en la vista, el controlador toma estos cambios detectados en la misma y realiza operaciones con el modelo (utilizando la lógica programada en sí), finalmente el modelo les avisa a las vistas que deben actualizarse con los nuevos datos.