

mat|r

Bienvenidos a la Clase N°01

En unos minutos comenzamos...

Clase 01

Introducción a Mat|r Script

Mat|r project

Ecosistema 360 para la creación de experiencias digitales

2020

Expositor



Agustina Dinamarca

Capacitadora mat|r

Se dedica a representar el producto en actividades que tienen como objetivo educar a usuarios sobre qué es mat|r, para qué sirve y cómo usarlo a través del dictado de webinars y capacitaciones, y el desarrollo de contenido educativo.

OBJETIVOS DE LA CLASE

- Acercar mat|r como herramienta novedosa para el desarrollo de aplicaciones.
- Conocer la plataforma y sus partes principales.
- Aprender funcionalidades básicas y conceptos elementales del lenguaje mat|r script.

HOY APRENDERÁS

- Qué es Mat|r, para qué sirve y qué propone.
- Sus beneficios y sus ventajas frente a otras tecnologías similares.
- Introducción a un nuevo lenguaje de desarrollo: mat|r script.
- La plataforma y sus componentes principales.
- Funcionalidades básicas.
- Conceptos elementales de mat|r script: modelos y experiencias.



¿QUÉ ES MATIR?

Es un ecosistema para el desarrollo de aplicaciones móviles.

Es una nueva forma de desarrollar aplicaciones que combina **pocas líneas de código, colaboración y reutilización en un entorno muy fácil de aprender y de usar**, permitiendo crear aplicaciones en una fracción de tiempo de lo que llevaría utilizar lenguajes tradicionales. Multiplataforma, sin necesidad de instalar nada, sin necesidad de configurar nada.





¿QUÉ PROPONE
MATIR?

¿QUÉ PROPONE MAT|R?

- Propone un lenguaje de programación agnóstico del dispositivo: **mat|r script**.
- El cambio paradigmático es programar directamente experiencias de usuario (UX), automatizando el enlace ("*binding*") al modelo de datos.
- Agilizar el tiempo de desarrollo.
- Un *Hub* de librerías y aplicaciones disponibles para la comunidad.





BENEFICIOS

BENEFICIOS



Cuando un modelo en matr es cambiado todas las apps generadas por ese modelo se **actualizan** instantáneamente. Olvida los largos tiempos de implementación de cambios y mejoras.

matr habilita la mejora continua con rápida respuesta a las necesidades de usuarios.

1

Colaborativo

2

Fácil de usar

3

Reducción en tiempo
de desarrollo

4

Multiplataforma

5

Fácil de
aprender

6

Fácil integración
con servicios

7

Gratis

8

Plataforma web



DIFERENCIA

VENTAJAS MAT|R VS. SOLUCIONES ACTUALES



- Poder de programación con la facilidad de uso de bajo código.
- No solo aplicaciones móviles: web, IoT, texto, voz, VR, AR, etc.
- Potente manejo de datos y actualizaciones de aplicaciones automáticas.
- *Backends* autogenerados: Google Firebase, AWS, Azure, SOLID.
- Despliegue o implementación en un click para todas las plataformas y dispositivos.
- Compartición y colaboración en tiempo real y a escala global.





PRINCIPALES COMPONENTES

MAT|R SCRIPT

ES FÁCIL CREAR EXPERIENCIAS AGNÓSTICAS DEL DISPOSITIVO

Una **Experiencia** genera un modelo y una UI asociada al mismo, con un enlace automático y bidireccional (*two-way data binding*) por defecto.

```
Experience Person {
    String name
    String lastname
    Integer age
    Date birthday
    Array<Hobbies> hobbies
}

Model Hobbies {
    String name
}
```

```
Rule birthday listen(birthday from Person as b) when(p.birthday == today()) {
    // code here...
}
```

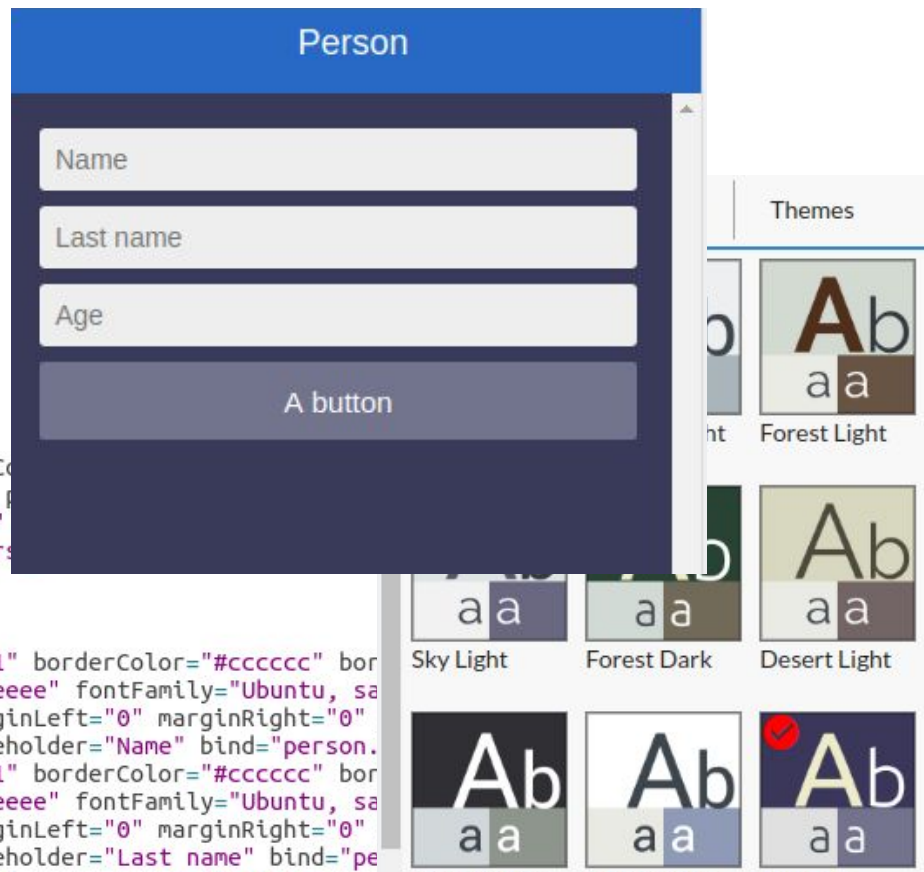

MAT|R UI BUILDER

LA UI ES AUTOMÁTICAMENTE GENERADA A PARTIR DE EXPERIENCIAS

Permite crear y actualizar automáticamente representaciones visuales (*layouts*).

Lenguaje de marcado
(*markup language*): XML.

```
<layout alpha="1" backgroundColor="white"
paddingBottom="18" paddingLeft="18" paddingRight="18"
paddingTop="18">
  <datasource type="Person">
    <navigationBar title="Person">
      <leftButtons/>
      <rightButtons/>
    </navigationBar>
    <textfield wpercentage="1" borderColor="#cccccc"
borderColor="#cccccc" backgroundColor="#e0e0e0"
fontFamily="Ubuntu, sans-serif"
marginBottom="8" marginLeft="0" marginRight="0"
paddingLeft="8" placeholder="Name" bind="person.name">
    <textfield wpercentage="1" borderColor="#cccccc"
borderColor="#cccccc" backgroundColor="#e0e0e0"
fontFamily="Ubuntu, sans-serif"
marginBottom="8" marginLeft="0" marginRight="0"
paddingLeft="8" placeholder="Last name" bind="person.lastName">
    <textfield wpercentage="1" borderColor="#cccccc"
borderColor="#cccccc" backgroundColor="#e0e0e0"
fontFamily="Ubuntu, sans-serif"
marginBottom="8" marginLeft="0" marginRight="0"
paddingLeft="8" placeholder="Age" bind="person.age">
    <button text="A button" bind="person.button">
  </layout>
```



MAT|R HUB

Mat|r fue diseñado desde su concepción para la colaboración: no necesitas escribir todo desde cero. En mat|r Hub encontrarás un centro de desarrollo social completo.

Concentra todas las aplicaciones y librerías publicadas por la comunidad.

Permite buscar, copiar, y ejecutar aplicaciones mat|r y todas las librerías que necesitas para **acelerar el desarrollo.**

The screenshot displays the Mat|r Hub interface. On the left is a dark blue vertical sidebar with icons for home, folder, share, chat, and a green speech bubble. The main content area is titled 'Hub' and has tabs for 'Applications' and 'Libraries'. It includes filters for 'All', 'Category', and 'Tags'. A grid of application cards is shown, each with a title, description, a lock icon, the word 'Public', and a star rating. The visible cards are:

- Prueb**: Codigo de Prueba Primer Intento Con Matr. Rating: 5 stars.
- fjexfgnsf**: asfdasf. Rating: 5 stars.
- Sample App**: Una prueba de apli. Rating: 5 stars.
- BitcoinPrice**: Shows current bitcoin price. Rating: 5 stars.
- Blue Now**: Blue Now App. Rating: 5 stars.
- BotControllo**: An application to h. Rating: 5 stars.
- Clase 2 - Curso m...**: Clase 2 - Curso mat|r. Rating: 5 stars.
- Clase3Example**: Clase 3. Rating: 5 stars.
- Clase 4 curs**: curso mat|r 4. Rating: 5 stars.

MAT|R VIEWER APP

TODO LO QUE DESARROLLES
PODRÁS VERLO FUNCIONANDO
INSTANTÁNEAMENTE EN LAS
PLATAFORMAS COMPATIBLES

Actualmente, mat|r es soportado por las dos principales plataformas móviles **iOS y Android**. Web y otros entornos están por venir y muchos más serán agregados por la comunidad y el equipo mat|r.



Puedes bajar mat|r viewer para probar tus aplicaciones mat|r ya mismo.

Para usar la app necesitan una cuenta en [mat|r project](#).

TEST QR

ESCANEA EL CÓDIGO QR CON TU DISPOSITIVO MÓVIL.

Instantáneamente transferirás tu mat|r app a tu dispositivo mobile :)

Comparte tu mat|r app.

Scan the QR to open your App!



Share your App!

1. Share the code with your collaborators.

Copy and share the code with your collaborators

GENERATE CODE



Para escanear el código QR recuerda que necesitarás tener la mat|r viewer app instalada en tu dispositivo.

2. Invite with email

Send the invitation code by email for collaborators can use your Mat|r app on their smartphones

Add sharing emails

Press<Enter>to confirm or add multiple emails

SEND

MAT|R DOCS

DOCUMENTACIÓN COMPLETA

En mat|r Docs encontrarás toda la información que necesitas sobre las distintas funcionalidades que ofrece mat|r, así como tutoriales para que empieces a utilizar nuestra herramienta.



Acerca de Mat|r

PLATAFORMA



Comenzando

TUTORIALES



Nuestro Lenguaje

MAT|R| SCRIPT



Nuestra Herramienta

UI-BUILDER

OTROS RECURSOS

MAT|R FORO

Conéctate con la comunidad Matr, haz preguntas y comparte experiencias.

APRENDE MAS →

WEBINARS

Videos de Youtube sobre funcionalidades Mat|r

APRENDE MAS →

MAT|R ACADEMY

El mejor lugar para comenzar a aprender y desarrollar tu primer aplicación

APRENDE MAS →

PREGUNTAS FRECUENTES

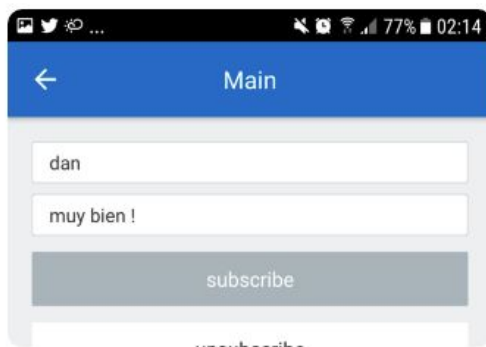
Todo lo que necesitas saber

APRENDE MAS →

OTROS RECURSOS

Mat|r|Blog

#TUTORIALES



Tus primeros pasos con MQTT en Mat|r

Autora: Ing. Ana Laura Diedrichs Intro Si ya creaste tu cuenta en Mat|r puedes empezar a probar...

```

array<Player> all = broker.cloudPersistence.get('Player')
if (all.size() < 2 || all.size() > 2) {
  for (one in all) {
    broker.cloudPersistence.remove(one)
  }
}

getIn = GetIn()
currentGame = Game()

broker.ui.push('Menu')
localPlayer = Player(nick: '')
ready_msg = Ready()

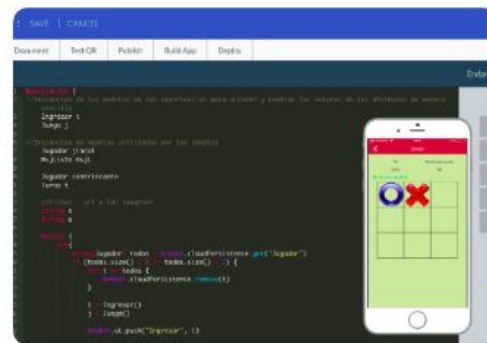
opponent = Player()
currentTurn = Turn(number: -1)

pieceUrl1 = 'http://matrproject.com/appimages/appposts/tateti-ui/x.pn
pieceUrl1 = 'http://matrproject.com/appimages/appposts/tateti-ui/o.pn

//async: inicializo y me suscribo
broker_async.initialize()
broker_async.connect()
  
```

Mat|r tips: Implementando la nueva UI a nuest...

Hola, soy Fernando nuevamente. En esta ocasión, y para complementar la guía anterior, vamos a retomar nuestra app...



Mat|r tips: Desarrollando un tateti

Hola, soy Fernando, especialista en Mat|r, y en esta ocasión vengo a ustedes para mostrarles algunas de las...

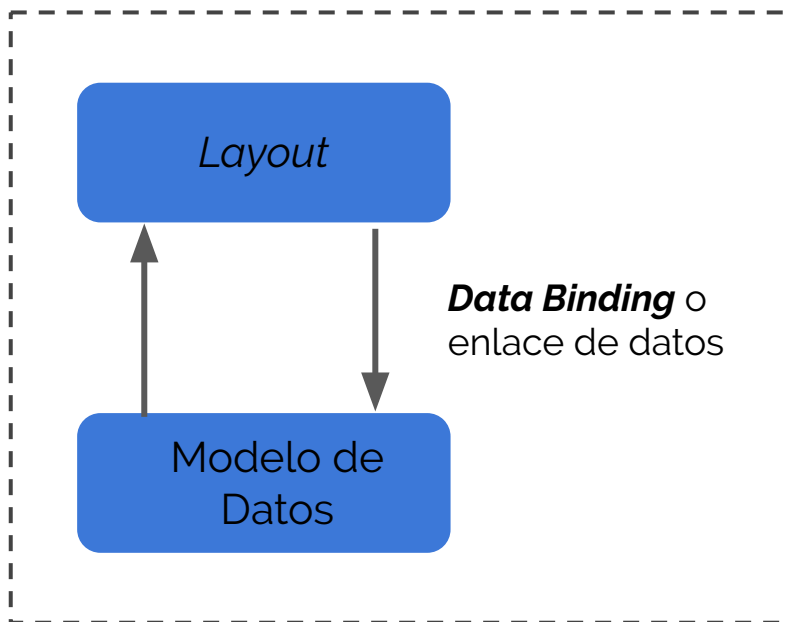


RECORRIDO POR LA PLATAFORMA (:



CONCEPTOS ELEMENTALES

CONCEPTOS ELEMENTALES



Experiencia

Nota:

layout = vista = representación visual

Sincronización **automática** y **bidireccional** de datos entre el modelo de datos y su representación visual.

CONCEPTOS ELEMENTALES

EXPERIENCIA

```

Experience Person {
  String name label("Name") as TextField
  String lastname label("Last name") as TextField
  Integer age label("Age") as TextField

  Decision btn action("ctx.crear") label("A
button")
}

```

LAYOUT

Binding

MODELO
DE
DATOS

```

Model Person {
  String name
  String lastname
  Integer age
}

Decision btn action("ctx.crear")
label("A button")

```

EJEMPLOS

¡importante! Crear un layout y/o añadir componentes visuales a uno existente no autogenera código en los archivos .ms

1. Crearemos una experiencia (= modelo de datos + representación visual o *layout*).
2. Crearemos directamente un *layout* y le asociaremos luego un modelo de datos estableciendo un enlace entre ellos (*binding*).

Veremos que frente a cambios en el modelo (o en el *layout*) se generan actualizaciones en el *layout* (o en el modelo). ¡Veamos cómo funciona esto!

main.ms x

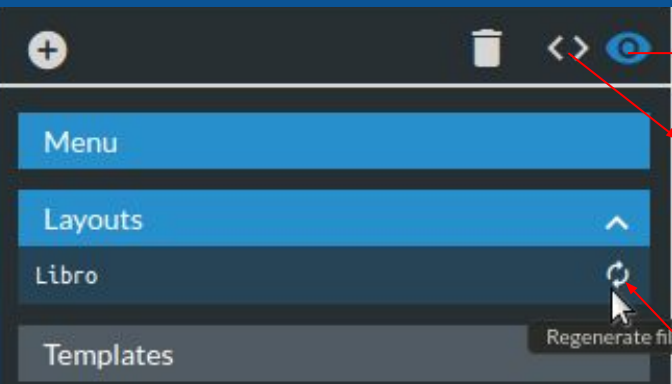
```
1
2 Experience Libro {
3     String titulo label("Título") as TextField
4     String autor label("Autor") as TextField
5     Integer ano label("Año") as TextField
6     String ubicacion label("Ubicación") // enlace a un componente visual TextField por default
7     // Defino un botón y le asigno una acción a ejecutar y un rótulo "Añadir"
8     Decision ingresar action("btnCtx.ingresar") label("Añadir")
9
10 }
11
12 RuleContext btnCtx {
13     Rule ingresar {
14         // bloque de código a ejecutar cuando ocurra el evento presionar botón
15     }
16 }
```

Atributos de la experiencia bindeados usualmente a componentes visuales del layout

Define un botón

Regla "ingresar" perteneciente al contexto de reglas "btnCtx" definida en otra parte del código

Definimos una experiencia "Libro" en el archivo "main.ms" en la sección "Model" del CDE.



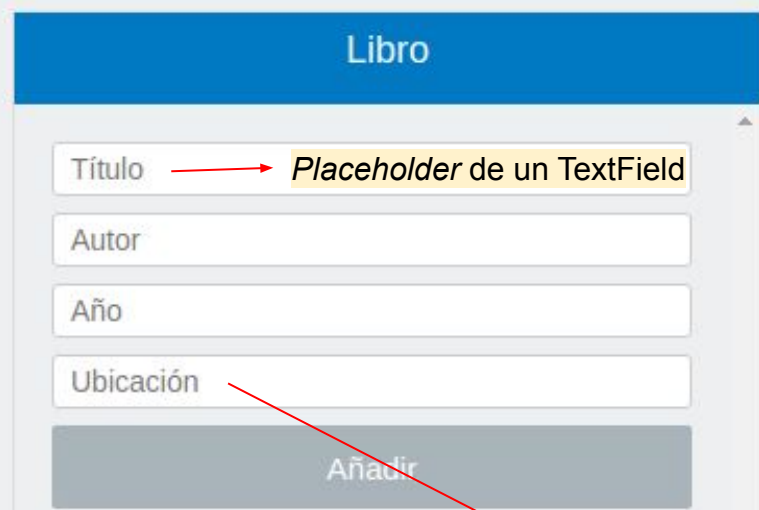
Vista gráfica

Vista en *markup language xml*

¡Ojo! Resetea **layout de una experiencia** según el modelo de datos y estilos visuales por default

- Layout
 - Textfield
 - Textfield
 - Textfield
 - Textfield
 - Button

Tipos de componentes visuales del layout "Libro"



Notar que el atributo "ubicacion" en la experiencia "Libro" no fue bindeado a ningún componente visual explícitamente (aparece por default asociado a un TextField)

Layout autogenerado y asociado a la experiencia "Libro" en la sección "Build UI" del CDE.

+

- Create menu
- Create layout
- Create template

Templates

Layout

- Textfield
- Textfield
- Textfield
- Textfield
- Button

Puedo crear un layout desde cero...

Libro

+

☒ <> 🔍

Menu

Layouts ^

Libro ↻

Templates

Layout

- ☐ Textfield
- ☐ Textfield
- ☐ Textfield
- ☐ Textfield
- ☐ Button

Libro

Create a new layout

Provide the new layout name.

Pelicula|

Layout de nombre "Pelicula"

Cancel Create

The screenshot shows the left sidebar of a development tool. At the top, there are icons for a plus sign, a trash can, a code editor, and a search icon. Below these are several menu items: 'Menu', 'Layouts', 'Libro', 'Película', 'Templates', and 'Layout'. The 'Película' item is highlighted with a red box, and a red arrow points from a text box to it. The 'Layouts' item has an upward arrow, 'Libro' has a refresh icon, and 'Película' has a downward arrow.

Película

El layout "Película" no proviene de haber definido una experiencia

Layout vacío, sin componentes visuales por el momento

The sidebar contains a list of components under the 'Layout' section:

- Menu
- Layouts
- Libro
- Pelicula
- Templates
- Layout
 - Label
 - Textfield
 - Textfield
 - Textfield
 - Button

Componentes visuales añadidos desde el "Palette"

The app form is titled "Pelicula" and contains the following elements:

- Header: Pelicula
- Text: Ingrese datos de una película:
- Textfield: Título
- Textfield: Director
- Textfield: Año
- Button: Añadir

Se dio un estilo predefinido desde "Themes"

Se modificó el texto en los componentes desde "Properties"

The right sidebar shows theme options under the 'Themes' tab:

Properties	Palette	Themes
Neon Dark	Neutral Light	Forest Light
Sky Light	Forest Dark	Desert Light
Neutral Dark	White Light	Sky Dark



main.ms
Libro.ui
Pelicula.ui

main.ms x

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

Modelo de datos a bindear al layout "Pelicula"
creado desde el Build UI.

```
Model pelicula {  
    String titulo  
    String director  
    Integer ano  
}  
  
RuleContext myCtx {  
    Rule anadirPelicula {  
        // bloque de código a ejecutar cuando ocurra el evento de presionar botón  
    }  
}
```

COMUNÍCATE CON NOSOTROS



support@matrproject.com



<http://forum.matrproject.com>

mat|r

matrproject.com

—

Contacto

Soledad Peñaloza

Community Manager

soledad.p@matrproject.com

Sergio Farias

Product Development

sergio.f@matrproject.com

Agustina Dinamarca

Capacitadora mat|r

agustina.d@matrproject.com

mat|r

iMuchas Gracias!

mat|r project

Ecosistema 360 para la creación de experiencias digitales