

Universidad Nacional Autónoma de México
Secretaría de Desarrollo Institucional

Dirección General de Cómputo y de
Tecnologías de Información y Comunicación



Software Low–Code Application Platform (LCAP) “Appian”

Dirección de Colaboración y Vinculación
Subdirección de Sistemas Integrados

Elaborado por: Wendy Hernández Méndez

Revisado por:

L.I. Luz María Ramírez Romero

L.I. Maribel Carmona Herrera

13 de diciembre de 2021

Índice

Introducción	4
Objetivo	5
Beneficios	5
Contenido	5
Appian	5
Costos:	5
Comunidad Appian	6
Prototipo desarrollado en Appian	6
Registro en la Comunidad Appian	6
Appian Console	7
Appian Designer	8
Appian RPA	9
Cloud Database	9
Tempo	10
Desarrollo del Prototipo	11
Group	12
Interface	13
Design Mode	15
Edit	15
Preview	16
Performance	16
Componentes de tipo Layouts.	21
Componentes de tipo Inputs.	21
Componentes de tipo Selection.	22
Desarrollo de una interfaz para un prototipo.	22
Rule Inputs	22
Agregando Componentes al Formulario.	23
Preguntas en un Formulario.	24
Bases de Datos	26
Data Type	26
Data Store	28
Constants	29

Process Model	31
Appian Process Modeler	32
Creación de Reportes con gráficas en Appian.	44
Record Type	44
Conectar una base de datos PostgreSQL	54
Resultados	58
Conclusiones:	58
Bibliografía/Referencias	58
Cursos de Appian consultados para el prototipo:	60
Glosario	60
Créditos	61

INTRODUCCIÓN

En este documento analizaremos el Low-Code Application Platform (LCAP) y ejemplificaremos su uso implementando un prototipo con la herramienta Appian.

El low-code es un término utilizado para hacer referencia al desarrollo de aplicaciones sin código o con poca construcción de código. En el caso concreto de que la aplicación no requiera escribir código, el software se denomina plataforma no-code. Este nuevo enfoque permite crear aplicaciones de negocio de forma rápida y sin necesidad de un alto conocimiento en TI.

Por otra parte, una plataforma de desarrollo low-code, es una herramienta que ayuda a crear visualmente aplicaciones completas utilizando una interfaz de arrastrar y soltar elementos.

Existen distintas herramientas de low-code que se pueden usar dependiendo del perfil:

- Desarrollador ciudadano: usan plataformas no-code o sin código.
- Desarrollador técnico no programador: analista de un negocio con conocimientos básicos en TIC.
- Desarrollador profesional: busca un entorno de desarrollo rentable.

Las plataformas de desarrollo low-code, generalmente contienen:

- Un entorno de desarrollo (IDE) visual: entorno para definir visualmente las interfaces de usuario, flujos de trabajo y los modelos de datos de su aplicación y, cuando sea necesario, añadir código escrito a mano.
- Conectores a varios back-ends o servicios: maneja automáticamente las estructuras de datos, el almacenamiento y la recuperación de los mismos.
- Gestor del ciclo de vida de aplicaciones: herramientas automatizadas para construir, depurar, desplegar y mantener la aplicación en prueba, puesta en marcha y producción.

Casos de uso:

Los casos de uso típicos para plataformas No-Code son los siguientes:

- Casos de negocio simples, que normalmente siguen el modelo CRUD (Crear, Leer, Actualizar y Eliminar).
- Reemplazar las hojas Excel que crean los usuarios en una compañía para complementar a los sistemas de información que disponen.
- Aplicaciones con un tiempo de vida corto.

- Procesos manuales que no requieren conexiones a sistemas terceros.

El caso de las plataformas Low-Code complementa a las No-Code con más características al permitir mayor control de la aplicación:

- Aplicaciones con un ciclo de vida mayor.
- Aplicaciones que se necesitan integrar con sistemas corporativos más complejos.
- Aplicaciones que requieren funcionalidades más complejas y personalizadas.
- Aplicaciones que cubran procesos de negocio críticos.
- Desarrollo de MVP (Productos Mínimos Viables), permite ver el resultado con menor inversión.
- Desarrollo de prototipos de aplicaciones para procesos de negocio.

OBJETIVO

Conocer el uso de low-code, sus beneficios y desventajas en el desarrollo de software. Analizar y explicar el funcionamiento de Appian para el desarrollo de software con low-code.

BENEFICIOS

Dar a conocer el uso de low-code como herramienta de desarrollo de software para la comunidad UNAM. Orientar al desarrollador en sus primeros pasos usando esta herramienta.

CONTENIDO

APPIAN

Appian es una herramienta que ayuda a construir aplicaciones y workflows rápidamente con una plataforma low-code. Combina personas, tecnologías y datos en un único workflow. Contiene desarrollo visual y componentes reutilizables.

COSTOS:

Planes	Free Edición de la comunidad de Appian.	Application Con todas las características.	Platform Con todas las características.	Unlimited Con todas las características.
Precio	Gratis	Standard: \$70 USD	Cotización personalizada.	Cotización personalizada.

		Infrequent: \$9 USD Input-Only: \$2 USD por usuario / por mes	Precio por usuario con aplicaciones ilimitadas.	Precio por desarrollo con aplicaciones ilimitadas.
Usuarios	15	Mínimo 100, sin máximo		Ilimitado
Apoyo	Comunidad en línea de Appian	El soporte avanzado y empresarial incluye alta disponibilidad para Appian Cloud		

Tabla 1. Planes de pago de la plataforma de low-code de Appian.

COMUNIDAD APPIAN

La empresa Appian cuenta con una comunidad en la cual se puede interactuar con personas que estén desarrollando en su plataforma de low-code, además de que existe un foro de preguntas donde se resuelven dudas acerca del desarrollo con Appian. Brinda acceso a documentación de la plataforma y cursos para aprender a desarrollar en su plataforma de low-code.

Para pertenecer a la Comunidad Appian sólo es necesario registrarse en su página con cualquier correo electrónico, pertenecer a la comunidad Appian es totalmente gratis.

PROTOTIPO DESARROLLADO EN APPIAN

A continuación, describiremos el proceso para crear un prototipo sencillo que contenga un formulario dirigido a la comunidad de Pre-becarios de la DGTIC además de reportes y gráficas analizando los datos recabados con el formulario.

REGISTRO EN LA COMUNIDAD APPIAN

Como vimos anteriormente para hacer uso gratuito de la herramienta de low code Appian es necesario crear una cuenta en la Comunidad de Appian.

Para registrarnos en la Comunidad Appian debemos de acceder al siguiente link:
<https://community.appian.com/>

Después de registrarnos, ingresamos a nuestra cuenta y en la siguiente opción creamos nuestro sitio.

My Community Edition Site

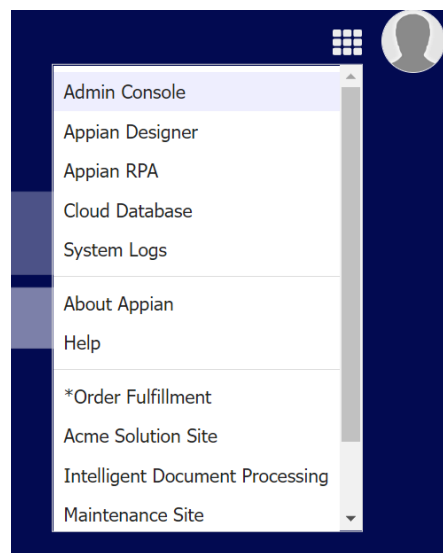
Community members get access to a free, unlimited-time Appian instance. Use it to get hands-on with Low-Code Automation.

[Get my site](#)

Después de crear el sitio llegará un correo con el link a tu sitio y sus credenciales. Solicitará cambio de credenciales. Después de eso puedes entrar a tu sitio para crear una aplicación.

Dentro de tu sitio se encuentran distintos módulos en el menú como:

- Admin Console
- Appian Designer
- Appian RPA
- Cloud Database
- Tempo



A continuación describiremos el propósito de algunos módulos con los que trabajaremos en el desarrollo de una aplicación.

APIAN CONSOLE

En este módulo puedes modificar características del sistema como el nombre del sitio, el logo, la dirección del sitio, los colores, entre otras características. Además de configuraciones de autenticación de usuarios.

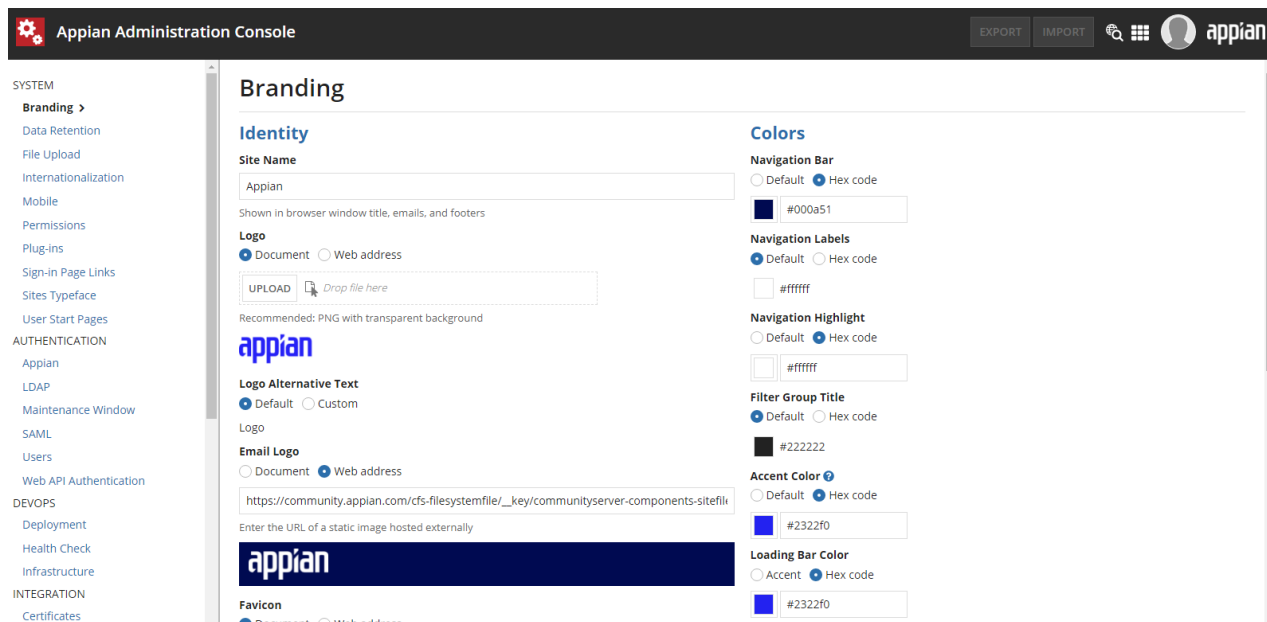


Imagen: Appian Administration Console.

APPIAN DESIGNER

Este módulo es el principal donde se desarrollan las aplicaciones. Cada elemento de la lista que se muestra en el inicio de Appian Designer es un ejemplo de una aplicación creada en Appian.

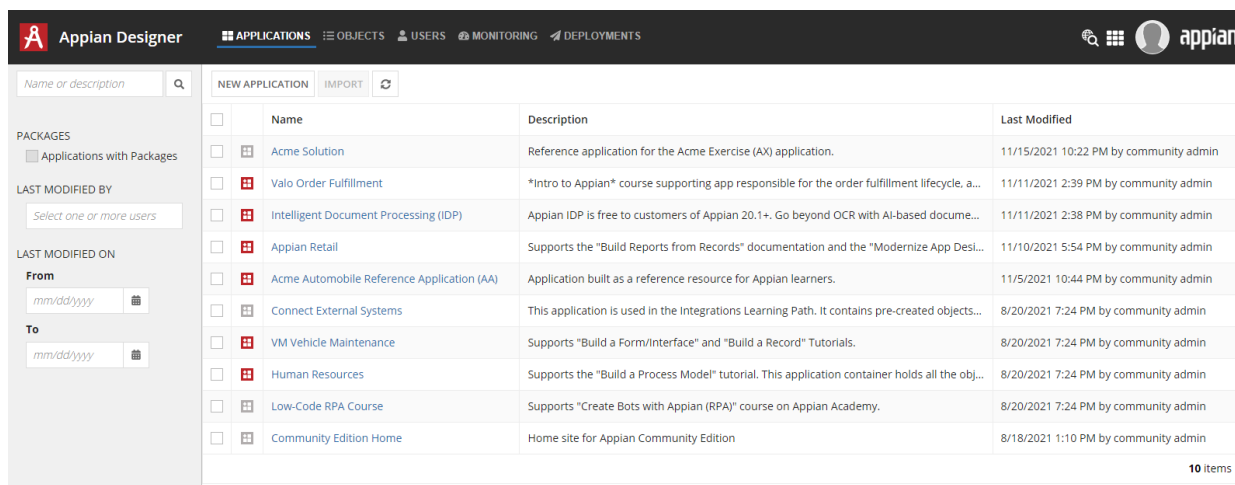


Imagen: Appian Designer.

APPIAN RPA

En el módulo Appian Robotic Process Automation (RPA) se pueden automatizar los procesos manuales de alto volumen y altamente transaccionales.

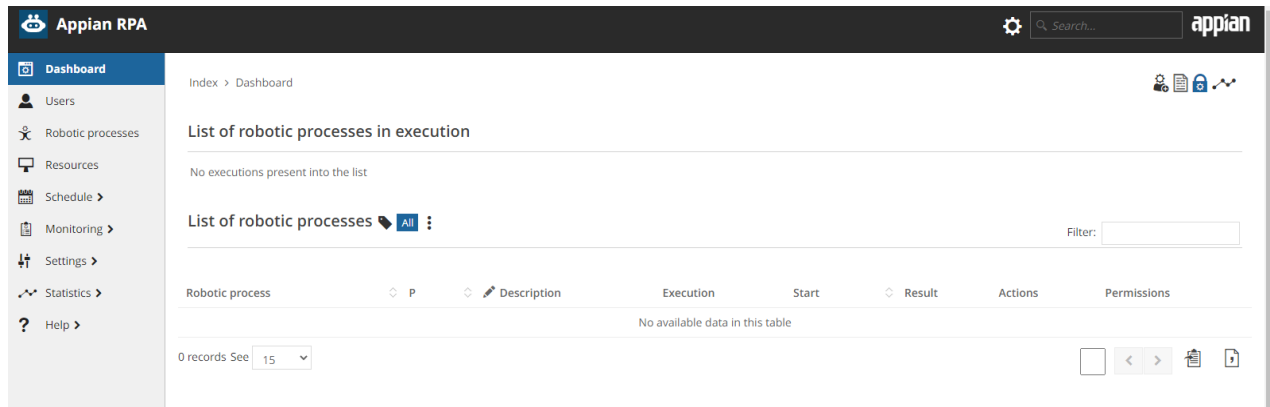


Imagen: Appian RPA.

CLOUD DATABASE

En este módulo se encuentra la base de datos de Appian. Appian Cloud está pre-configurada para poder concentrarse en crear aplicaciones de bajo código sin configurar ninguna base de datos.

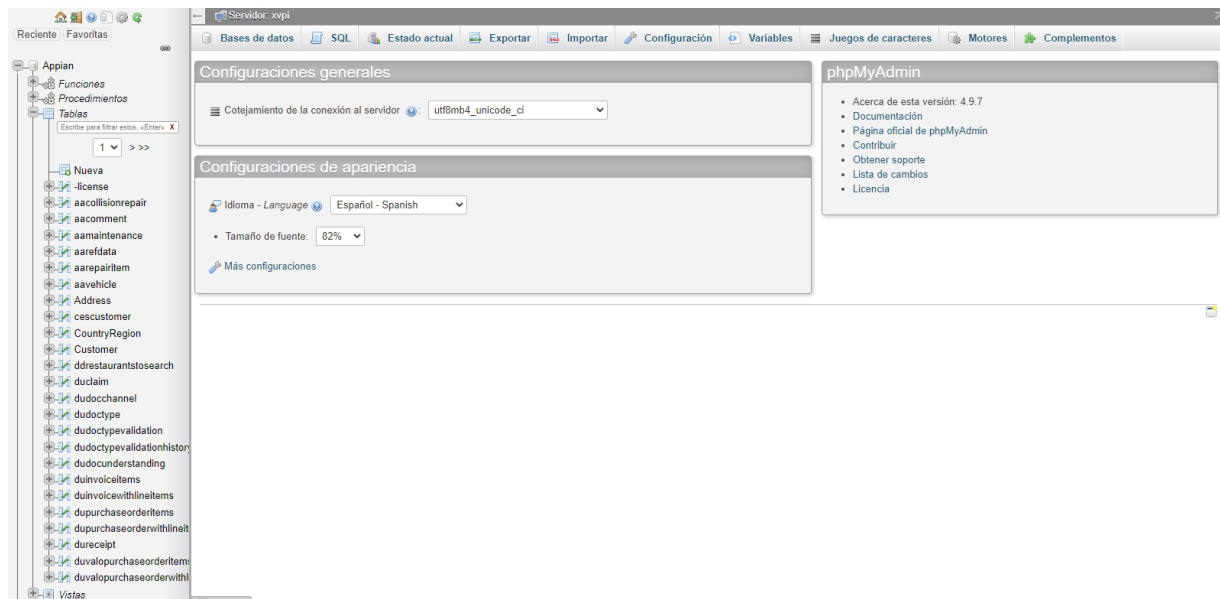


Imagen: Cloud Database.

TEMPO

Aquí los usuarios finales visualizan tu aplicación.

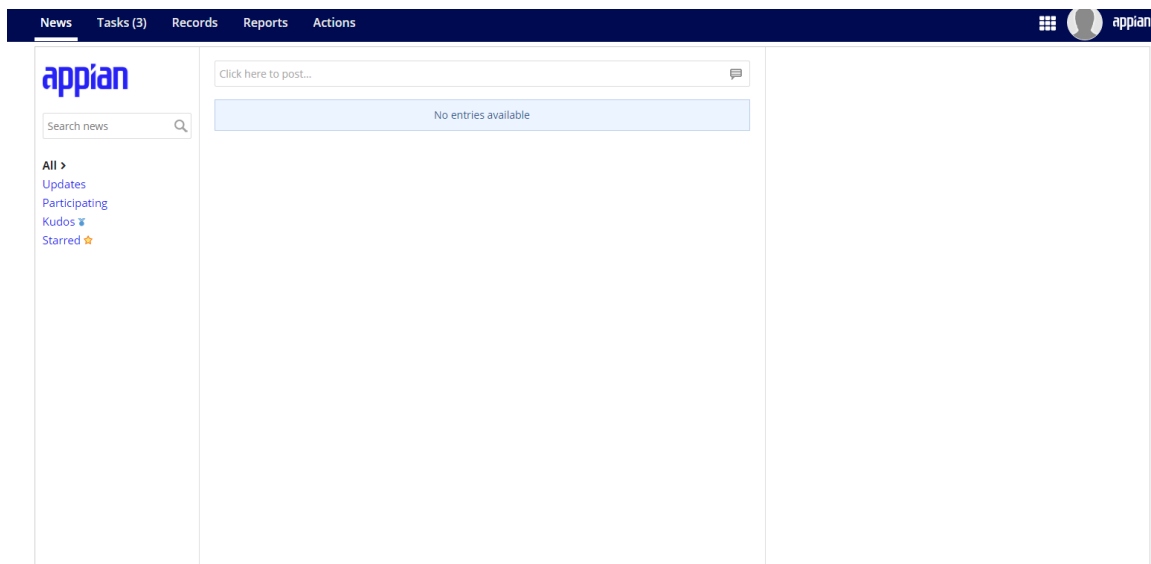
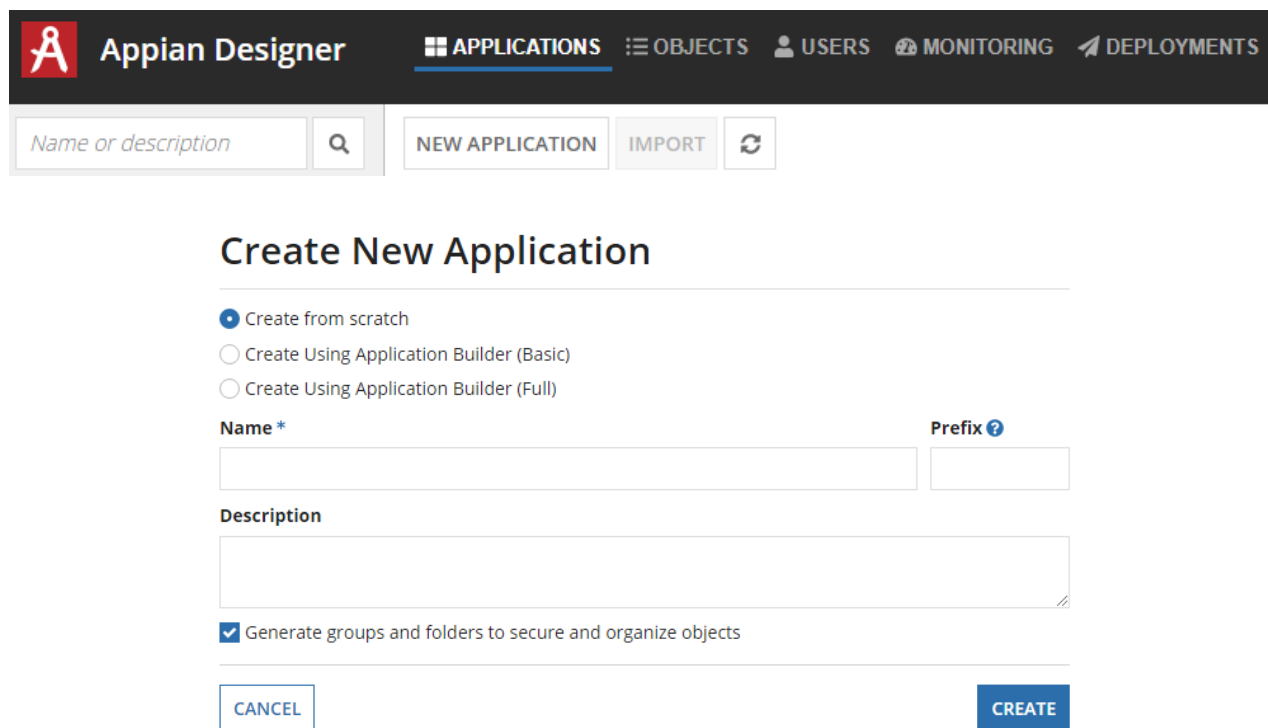


Imagen: Tempo.

DESARROLLO DEL PROTOTIPO

Comenzaremos creando una nueva aplicación en **Appian Designer**. Para crear una nueva aplicación es necesario seleccionar la opción de **NEW APPLICATION**.

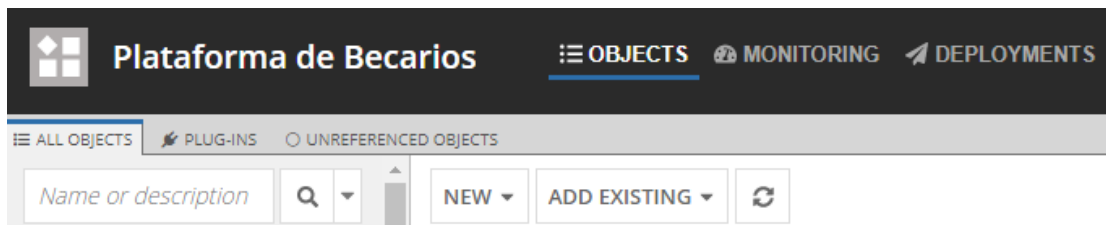


Es posible crear aplicaciones desde el inicio o crear aplicaciones con una estructura determinada.

Crearemos una aplicación desde el inicio.

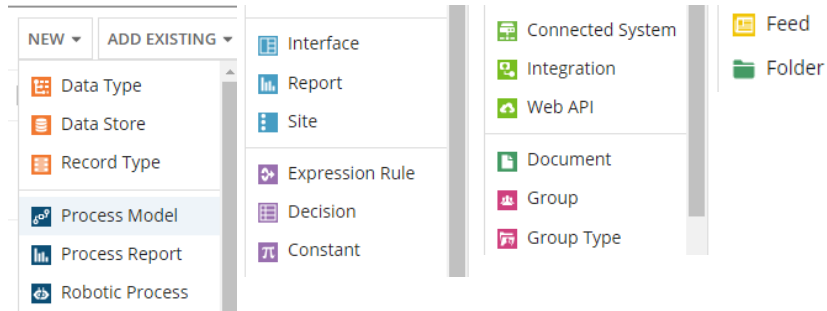
Al crearla **from scratch** se inicia la aplicación vacía. Es posible agregar una descripción de cada componente que se crea. Y al agregar un prefijo cada componente creado tendrá este prefijo para identificarlos de componentes de otras aplicaciones.

Después de crear la aplicación ingresamos a ella.



En la opción **NEW** puedes crear distintos elementos de tu aplicación como:

- Tipos de datos
- Modelos
- Reportes
- Constantes
- Grupos
- Interfaces



Todos son elementos necesarios para construir una aplicación dentro de Appian.

Group

Podemos crear Grupos dentro de nuestra aplicación que son equivalentes a crear roles en una aplicación.

Crearemos tres grupos: Administrador, Académico y Becario. Seleccionamos la opción de **Group** en **NEW** para crear un nuevo grupo.

Create Group

Name *

Description

Parent Group

Group Members

Group Type

Visibility

Privacy Policy

Al crear cualquier componente, se agregan configuraciones requeridas para cada uno. Puedes elegir la configuración que se adecue a las necesidades de tu aplicación. Para este ejemplo dejaremos las configuraciones más básicas.

Interface

Para crear una Interfaz necesitamos crear primero un **Folder** que sea de tipo **Rule Folder**. Por recomendaciones de Appian deberás nombrar a este folder **Rules and Constants**.

Create Folder

Type

- ☒ Rule Folder
☐ Process Model Folder
☐ Document Folder
☐ Knowledge Center

Folder Name ***Folder Description****Parent Folder**

Después creamos una nueva interfaz en la opción **Interface** de **NEW**.

Se pueden crear interfaces duplicando las ya existentes, de esta manera obtienes Interfaces con el mismo formato. Es necesario guardar la interfaz en el folder **Rules and Constants**.

Create Interface

☒ Create from scratch ☐ Duplicate existing interface

Name *

PDB_Evaluacion

Description

Save In *

 PDB Rules and Constants 



[Create New Rule Folder](#)

CANCEL

CREATE

Después de crear una Interfaz se direcciona la página a **Design Mode**, donde aparece un video tutorial que muestra cómo usar esta herramienta.

Appian's Interface Design Experience

This short video covers everything you need to know to get started building interfaces with the speed and simplicity of drag and drop.



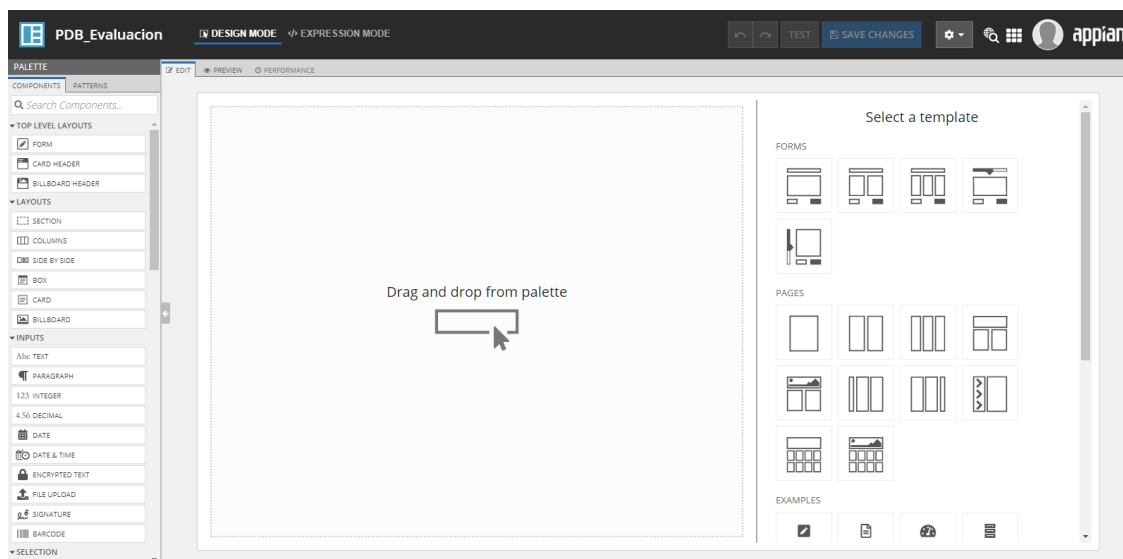
[See details on these features and more in our documentation](#)

☐ Don't show this dialog again

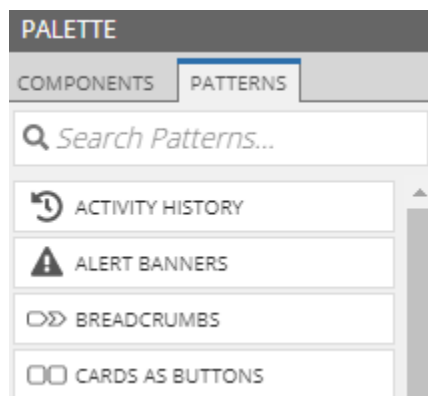
CLOSE

Design Mode

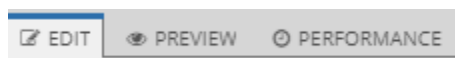
Design Mode es una herramienta de Appian que permite crear Interfaces con estructuras predeterminadas mediante drag and drop.



Contiene distintos tipos de componentes y patrones drag and drop en la sección **Palette**.



Existen tres formas de visualizar el desarrollo de tu aplicación: Edit, Preview y Performance.



Edit

En Edit se ve el posicionamiento de tus componentes dentro de cajas y señalamientos.



ELABORADO POR: **WENDY HERNÁNDEZ MENDOZA**

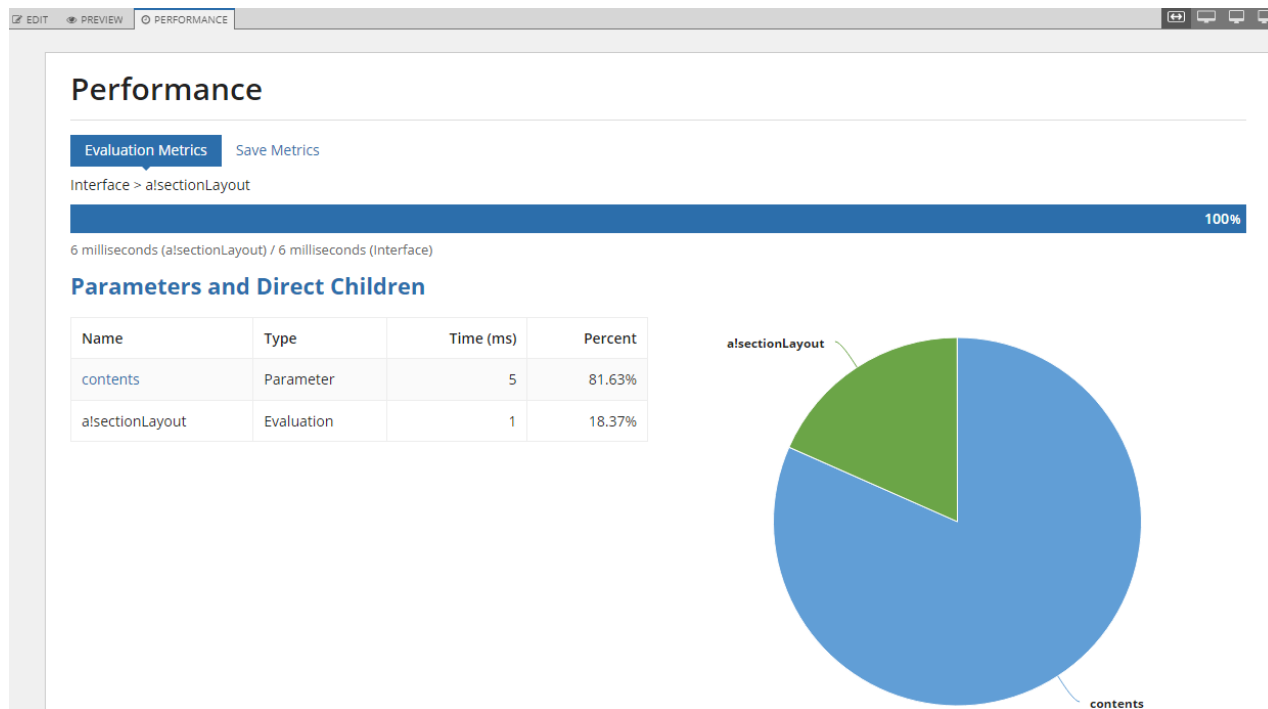
REVISADO POR: L.I. LUZ MARÍA RAMÍREZ ROMERO Y
L.I. MARIBEL CARMONA HERRERA

Preview

En preview se muestra la vista previa de la interfaz que se está desarrollando.

Performance

En performance se muestra un reporte de los componentes y campos que se utilizan en la interfaz que se está desarrollando.



Design Mode también cuenta con opciones para la visualización de la interfaz en distintos dispositivos como celulares, tablets, computadoras, etc.

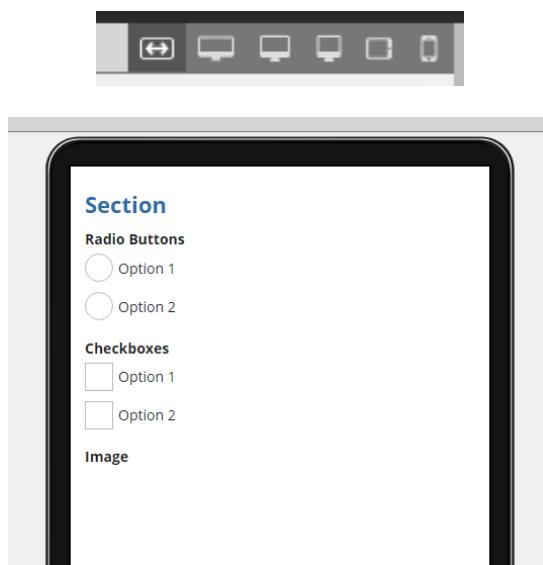
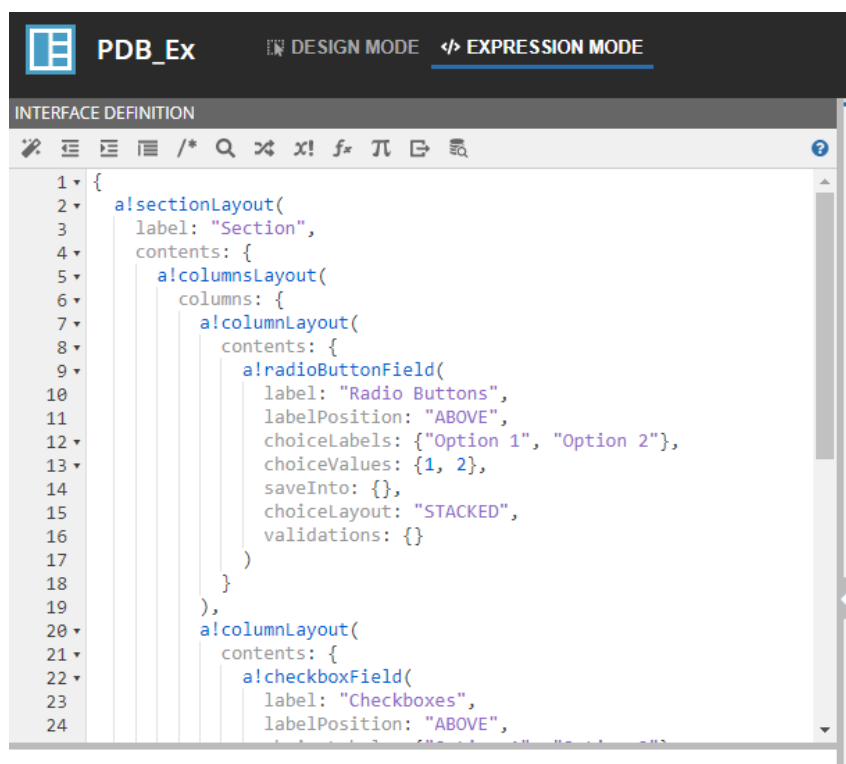


Imagen: Visualización de la Interfaz en un celular.

Además de contar con drag and drop, Design Mode cuenta con la opción **Expression Mode**. Característica que permite diseñar una interfaz con código.



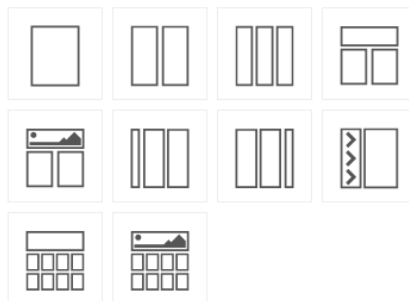
Design Mode permite seleccionar entre distintas plantillas para comenzar a crear tu interfaz.

Select a template

FORMS

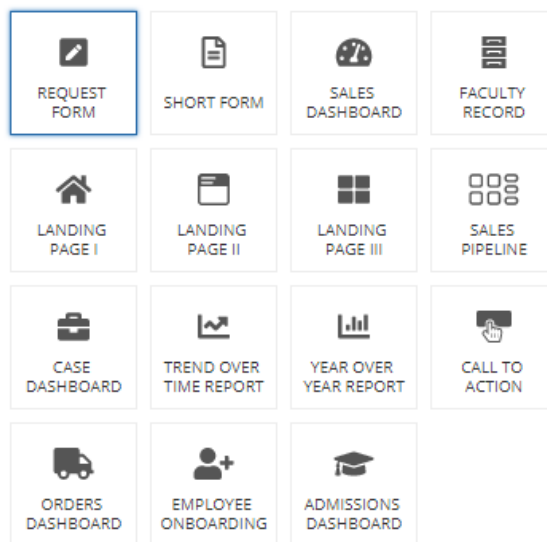


PAGES

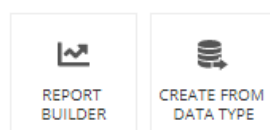


Además contiene distintos ejemplos de interfaces y permite crearlas desde bases de datos.

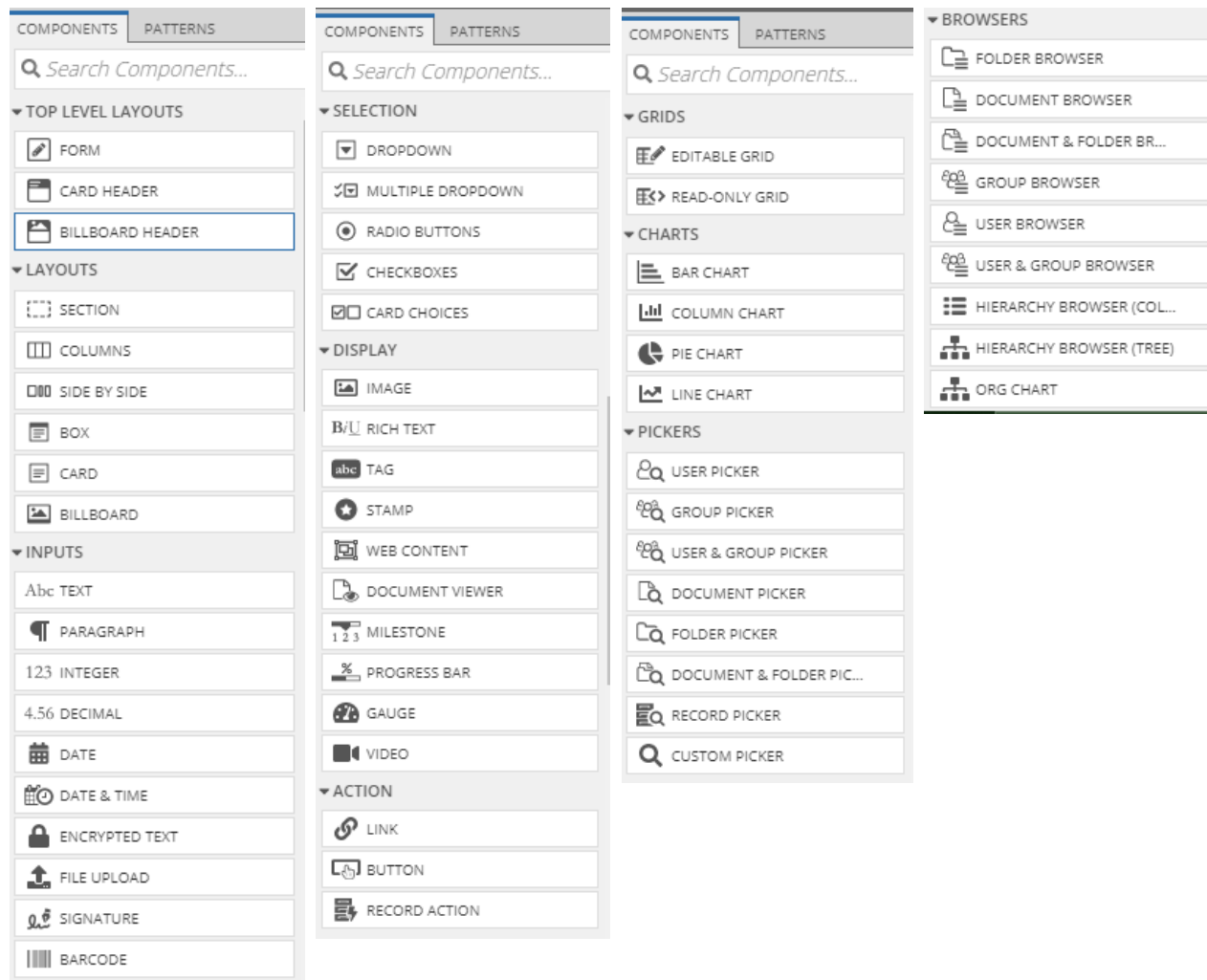
EXAMPLES



BUILDERS



Es posible ocupar distintos tipos de componentes para crear una interfaz.

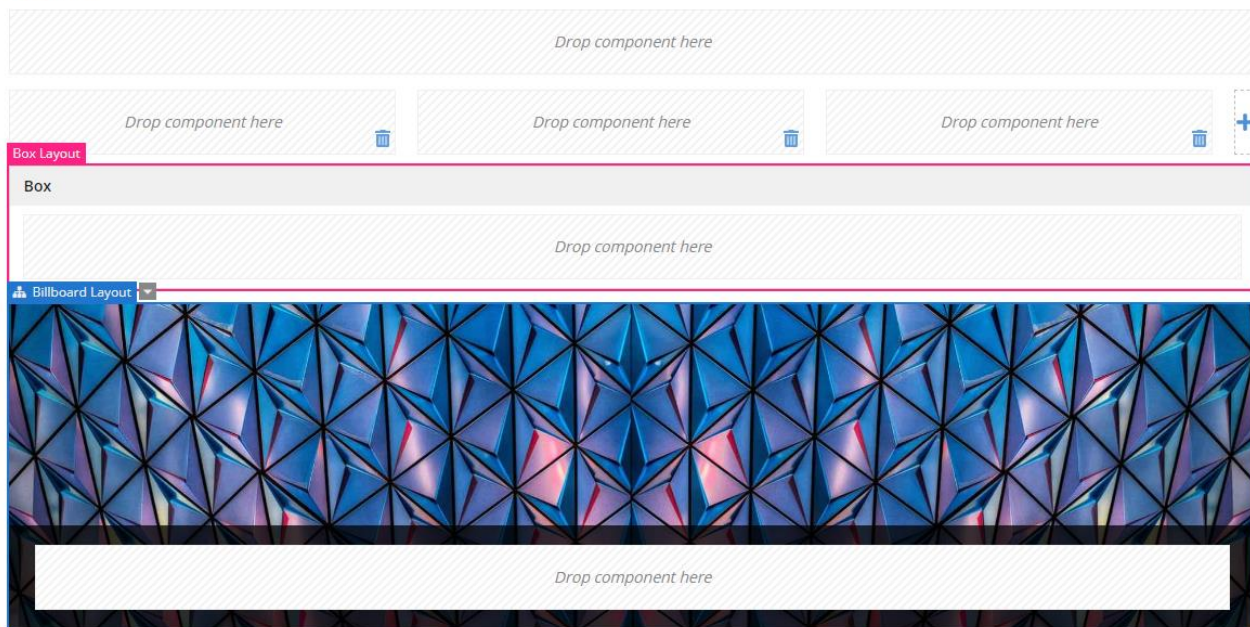


A continuación se visualizan algunos componentes disponibles en Design Mode.

Componentes de tipo Layouts.

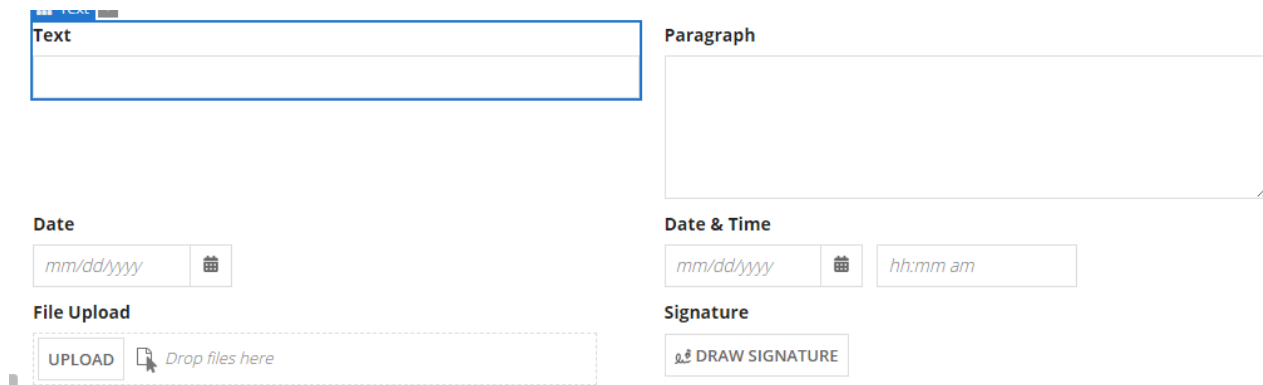
Visualización de componentes Sections, Columns, Box y Billboard.

Section



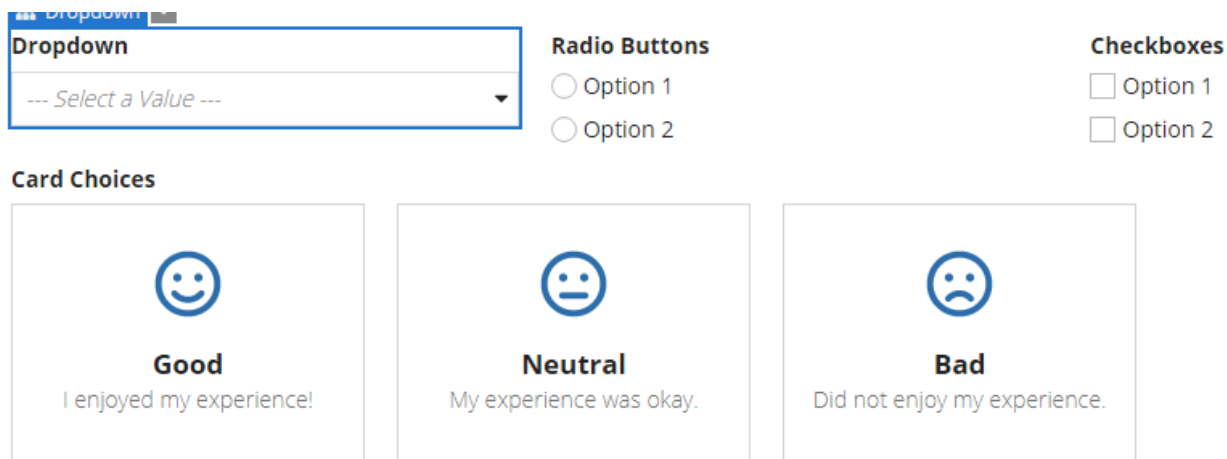
Componentes de tipo Inputs.

Visualización de componentes Text, Paragraph, Date, Date & Time, File Upload y Signature.



Componentes de tipo Selection.

Visualización de componentes Dropdown, Radio Buttons, Checkboxes y Card Choices.



The screenshot displays four types of selection components in Appian:

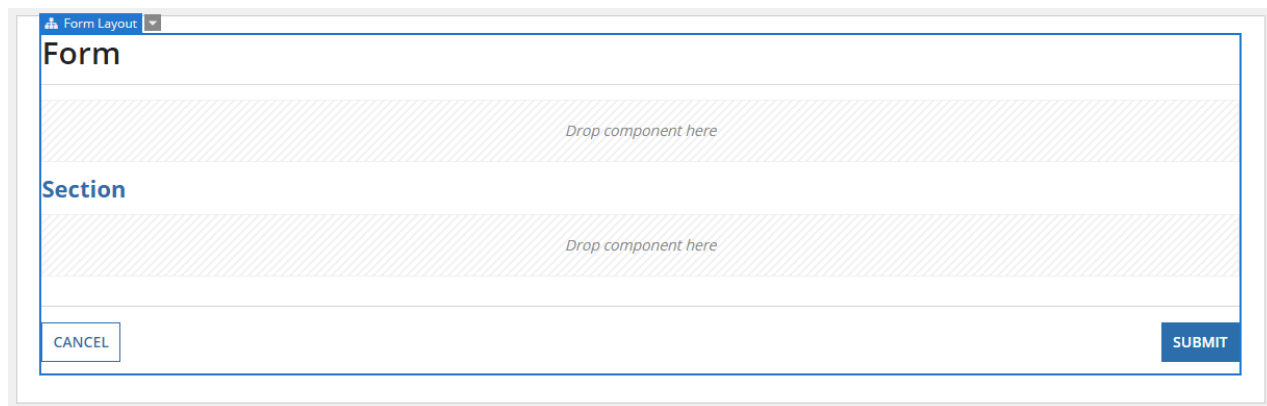
- Dropdown:** A component with a label "Dropdown" and a placeholder text "--- Select a Value ---".
- Radio Buttons:** Two radio button options labeled "Option 1" and "Option 2".
- Checkboxes:** Two checkbox options labeled "Option 1" and "Option 2".
- Card Choices:** Three cards representing experience levels:
 - Good:** Features a smiley face icon and the text "I enjoyed my experience!".
 - Neutral:** Features a neutral face icon and the text "My experience was okay."
 - Bad:** Features a frowny face icon and the text "Did not enjoy my experience."

Design mode contiene además componentes de tipo Display, Action, Grids, Chart, Pickers y Browsers.

Desarrollo de una interfaz para un prototipo.

Crearemos un formulario para documentar la Evaluación de Desempeño de Becarios.

Seleccionamos la plantilla de un **Form** básico. Observa que contiene un botón de cancelar y otro de enviar.



The screenshot shows a basic Appian Form template. It includes a title bar with "Form Layout" and a dropdown arrow. The main area is divided into two sections, each with a "Drop component here" placeholder. At the bottom, there are two buttons: "CANCEL" and "SUBMIT".

Rule Inputs

Para comenzar necesitamos crear **Rule Inputs**. Observa que al crear un formulario automáticamente se crea una Rule Input llamada **cancel**, esta contiene el valor del botón de Cancelar del formulario. Debes de crear una nueva Rule Input para los valores de las preguntas del formulario.

RULE INPUTS	
Name	Value
<input type="radio"/> cancel	false
<input type="radio"/> evaluacion	null

Agregando Componentes al Formulario.

Agregamos el nombre del formulario y seis componentes de **SECTION** para los distintos temas a evaluar: Asistencia, Actitud hacia las actividades encomendadas, Compromiso, Conocimiento en el área de cómputo, Calificación global y Comentarios.

Form Layout

EVALUACIÓN DEL DESEMPEÑO DE EL (LA) BECARIA(A)

Asistencia

Drop component here

Actitud hacia las actividades encomendadas

Drop component here

Compromiso

Drop component here

Conocimiento en el área de cómputo

Drop component here

Calificación global

Drop component here

Comentarios

Cada componente puede configurarse de acuerdo a las necesidades de la interfaz. Si agregas un componente a la interfaz del lado derecho de la pantalla se muestran las opciones de configuración del componente.

Cada componente tiene distintas opciones de configuración que van desde agregar el label, instrucciones, visibilidad del componente, si es un campo requerido, dónde guardar la información ingresada, entre otras opciones.

Form Layout

Section Layout

Label
EVALUACIÓN DEL DESEMPEÑO DE EL (

Instructions

Contents
[List of Components](#)

Buttons
[Button Layout](#)

Visibility
☒ Always show
☐ Only show when...

Validations
[List of Any Type](#)

Label CONFIGURATIONS

Label
Asistencia

Icon

Icon Alternative Text

Size
Default

Accessibility Heading Tag
Default

[Best Practices](#)

Color
☒ Choose a pre-defined color
☐ Use a custom color

Default

CONTENTS CONFIGURATIONS

Preguntas en un Formulario.

Para las preguntas agregamos componentes tipo **RADIO BUTTONS**.

EVALUACIÓN DEL DESEMPEÑO DE EL (LA) BECARIA(A)

Asistencia
☒ Radio Buttons

El(la) becario(a) participa:
☐ De forma regular
☐ Esporádicamente

La conexión del(la) becario(a) es:
☐ Puntual
☐ Impuntual

☒ Radio Buttons
Su participación en los cursos, actividades y proyecto es de:
☐ Menos de 4 horas diarias
☐ 4 horas diarias
☐ De 5 horas diarias o más
☐ Varía en días y horas

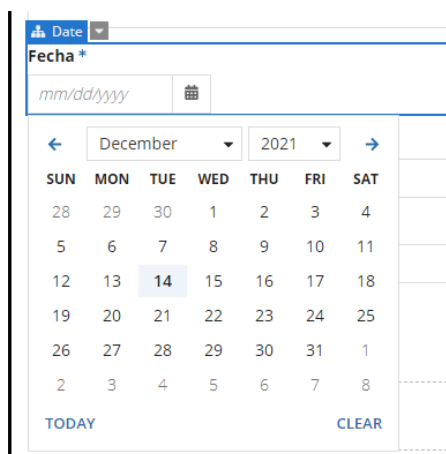
La participación del(la) becario(a) por lo regular:
☐ Es diligente y no le faltan ganas como se espera

Agregar cuadros de texto para los comentarios.

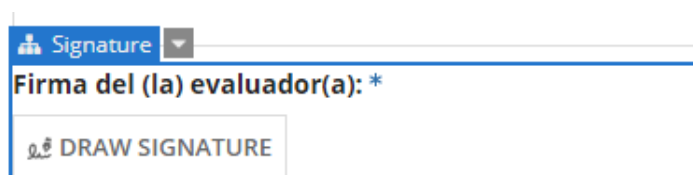
Comentarios

Qué otros comentarios y sugerencias daría para el mejor desempeño del(la) becario(a) dentro del proyecto:

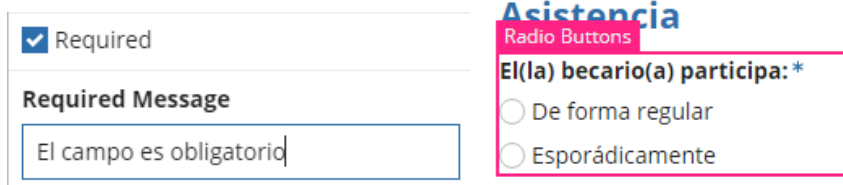
Y para seleccionar la fecha, dentro de un formulario, un componente tipo **Date**.



Es posible agregar campos para firmas, componentes tipo **Signature**. Este componente abre un cuadro para dibujar la firma y la guarda en formato PNG.



Es posible asignar campos requeridos y agregar mensajes en la configuración de cada componente. Al marcar un componente como requerido se agrega en el texto del Label un asterisco.



Con la información anterior es suficiente para crear la visualización completa de una interfaz, en este caso un formulario.

EVALUACIÓN DEL DESEMPEÑO DE EL (LA) BECARIO(A)

Datos del(la) becario(a)

Id del Becario *

Asistencia

El(la) becario(a) participa: *

☐ De forma regular

☐ Esporádicamente

La conexión del(la) becario(a) es: *

☐ Puntual

☐ Impuntual

Su participación en los cursos, actividades y proyecto es de: *

☐ Menos de 4 horas diarias

☐ 4 horas diarias

☐ De 5 horas diarias o más

☐ Varía en días y horas

La participación del(la) becario(a) por lo regular: *

Ahora es necesario agregar las funcionalidades y conectar el formulario con la base de datos para guardar la información ingresada en el formulario.

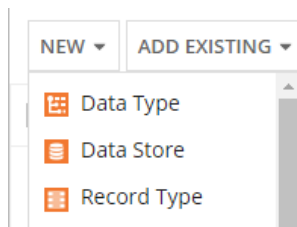
Bases de Datos

Appian permite la integración de fuentes de datos nativas, así como ingestión de datos desde hojas de cálculo, servicios web y otras fuentes. Puedes conectar tu aplicación con MySQL, MariaDB, Oracle, SQL Server, PostgreSQL y DB2.

Vamos a guardar los datos en la aplicación de Appian que utiliza MySQL.

Data Type

Para guardar los datos, es necesario crear tipos de datos (**Data Type**).



Create Data Type

☒ Create from scratch

☐ Duplicate existing data type

☐ Create from database table or view

☐ Import XSD

Namespace *

urn:com:appian:types:PDB

Formatted as a URI, for example 'urn:com:appian:types:COB' for a client onboarding application

Name *

PDB_

Description








CANCEL

CREATE

Puedes crear tipos de datos desde el inicio, duplicando tipos de datos existentes, desde tablas de bases de datos o importando tablas.

En estos Data Type creamos los atributos que contendrá la tabla de nuestra base de datos. Es posible configurarlos de distintos tipos, de distinto tamaño y agregar llaves primarias.

Fields (1)

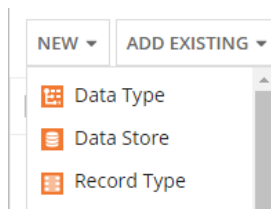
Name	Type	Length	Array	Key			
id_becario	 Number (Integer) 		<input type="checkbox"/>				
 New Field							

Los Data Type son equivalentes a tablas de una base de datos.

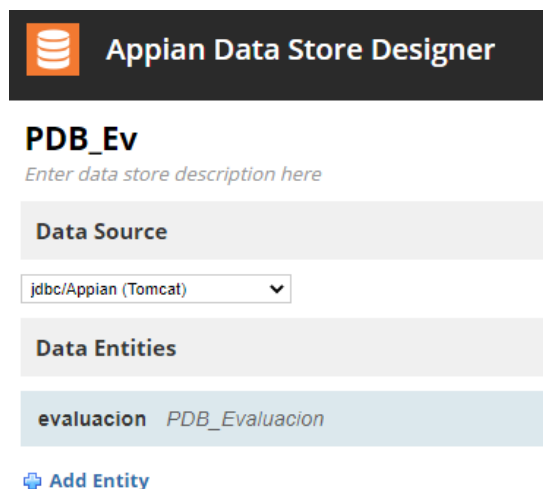
Para el formulario que estamos desarrollando agregaremos un atributo por cada pregunta, para guardar los datos ingresados en él.

Data Store

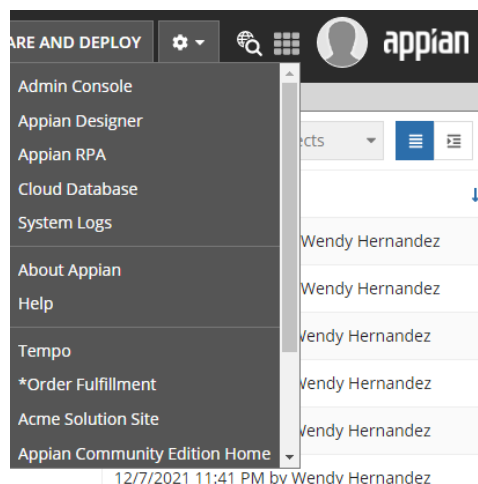
Después de guardar el Data Type con cada atributo necesario creamos una base de datos seleccionando la opción de **Data Store**.



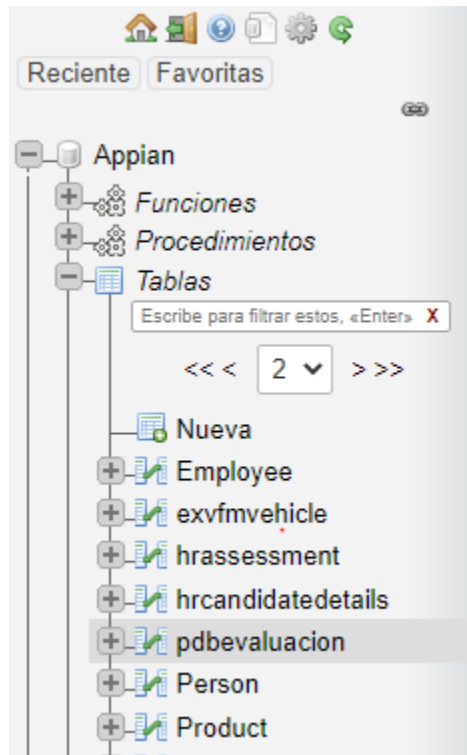
Creamos un Data Store y agregamos el Data Type que creamos en la opción de Data Entities.



Como ya vimos podemos ver las tablas del Data Store en la opción de Cloud Database, del menú de navegación.



Si ingresamos a Cloud Database podemos ver que nuestra tabla (*pdbevaluacion*) se creó correctamente.

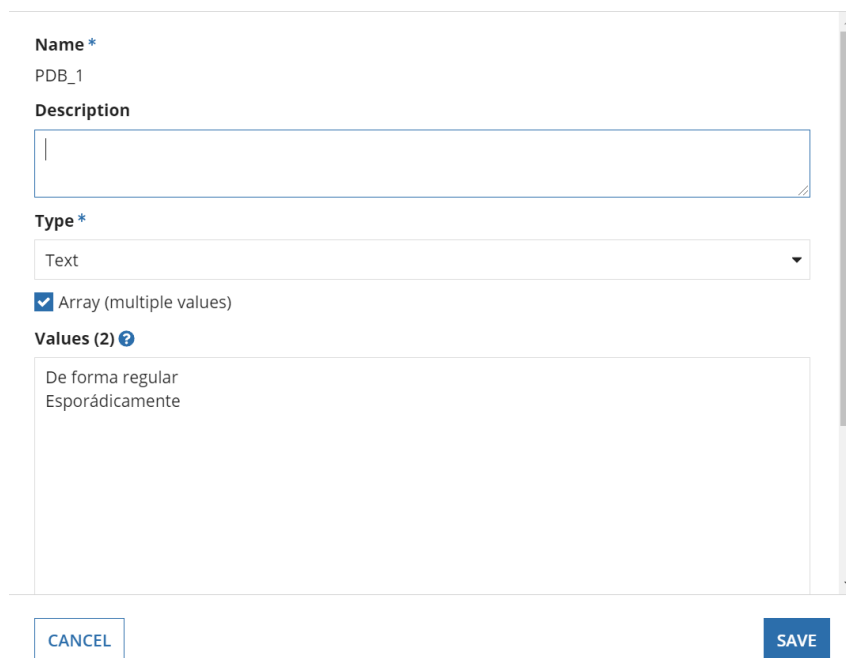


Ahora ya tenemos una base de datos para guardar la información del formulario pero para lograr esto aún se necesitan de varios elementos para su buen funcionamiento.

Constants

Para mostrar y guardar las opciones de Radio Buttons o Dropdown, entre otros. Hacemos uso de **Constants**. Debemos de crear una constante para cada Radio Button utilizado que contenga en un arreglo las opciones de respuesta a la pregunta formulada en el Radio Button.

Constant Properties



Name *
PDB_1

Description

Type *
Text

☒ Array (multiple values)

Values (2)

- De forma regular
- Esporádicamente

CANCEL SAVE

Marcamos la opción de **Array (multiple values)** y escribimos en **Values**, las opciones disponibles para la respuesta a la pregunta. Y las guardamos. Escribir una constante por pregunta si las opciones no se repiten.

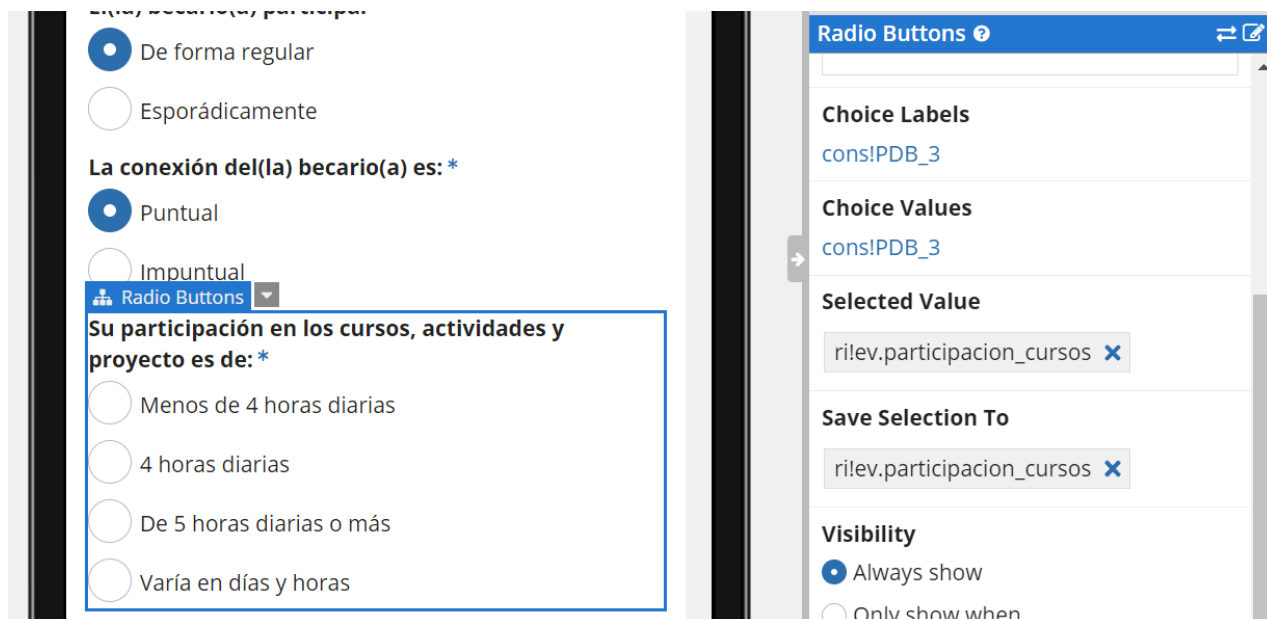
Observación: Si las opciones de respuesta a una pregunta se repiten en más preguntas puedes hacer uso de la misma constante.

Configurando Constants y Data Types en una interfaz.

Regresando a la interfaz del formulario en Design Mode podemos configurar las opciones de respuesta. Agregamos la **Constant** creada para cada pregunta en los valores de **Choice Labels** y **Choice Values**.

Y en las opciones de **Selected Value** y **Save Selection To** agregaremos el atributo del Data Type que creamos para cada pregunta.

En la imagen de abajo se ejemplifican estos valores.



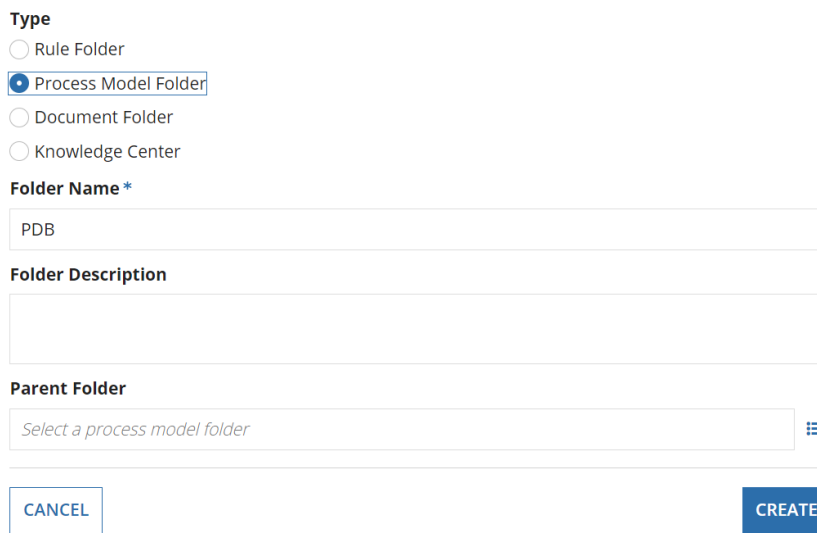
The screenshot shows a form configuration in Appian. On the left, a form section titled 'Radio Buttons' is visible, containing several radio button options. The first section has two options: 'De forma regular' (selected) and 'Esporádicamente'. The second section is titled 'La conexión del(la) becario(a) es: *' and has two options: 'Puntual' (selected) and 'Impuntual'. Below this, a dropdown menu is set to 'Radio Buttons'. The third section is titled 'Su participación en los cursos, actividades y proyecto es de: *' and has four options: 'Menos de 4 horas diarias', '4 horas diarias', 'De 5 horas diarias o más', and 'Varía en días y horas'. On the right, a configuration panel for 'Radio Buttons' is shown. It includes fields for 'Choice Labels' (cons!PDB_3), 'Choice Values' (cons!PDB_3), 'Selected Value' (rilev.participacion_cursos), 'Save Selection To' (rilev.participacion_cursos), and 'Visibility' (Always show).

Ya indicamos dónde se deben de guardar las respuestas al formulario en la base de datos, ahora debemos conectar la base de datos al formulario.

Process Model

Para conectar la base de datos a una interfaz se debe crear un Folder de tipo **Process Model Folder**.

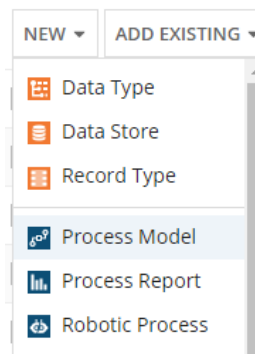
Create Folder



The screenshot shows the 'Create Folder' dialog in Appian. It has a 'Type' section with four radio button options: 'Rule Folder', 'Process Model Folder' (selected), 'Document Folder', and 'Knowledge Center'. Below this is a 'Folder Name *' field with the value 'PDB'. There is a 'Folder Description' field which is empty. Below that is a 'Parent Folder' field with the placeholder text 'Select a process model folder'. At the bottom, there are two buttons: 'CANCEL' and 'CREATE'.

Imagen: Creación de un folder de tipo Process Model Folder.

Y después crear un **Process Model** en el Folder anterior. En el Process Model se diseña el proceso que debe de seguir una aplicación desde que la inicias y además contiene las rutas que tomará la aplicación al presionar botones o al cancelar una operación.



Create Process Model



☒ Create from scratch ☐ Duplicate existing process model

Name *

PDB Prototipo

Description

Save In *

 PDB Process Models 



[Create New Process Model Folder](#)

CANCEL

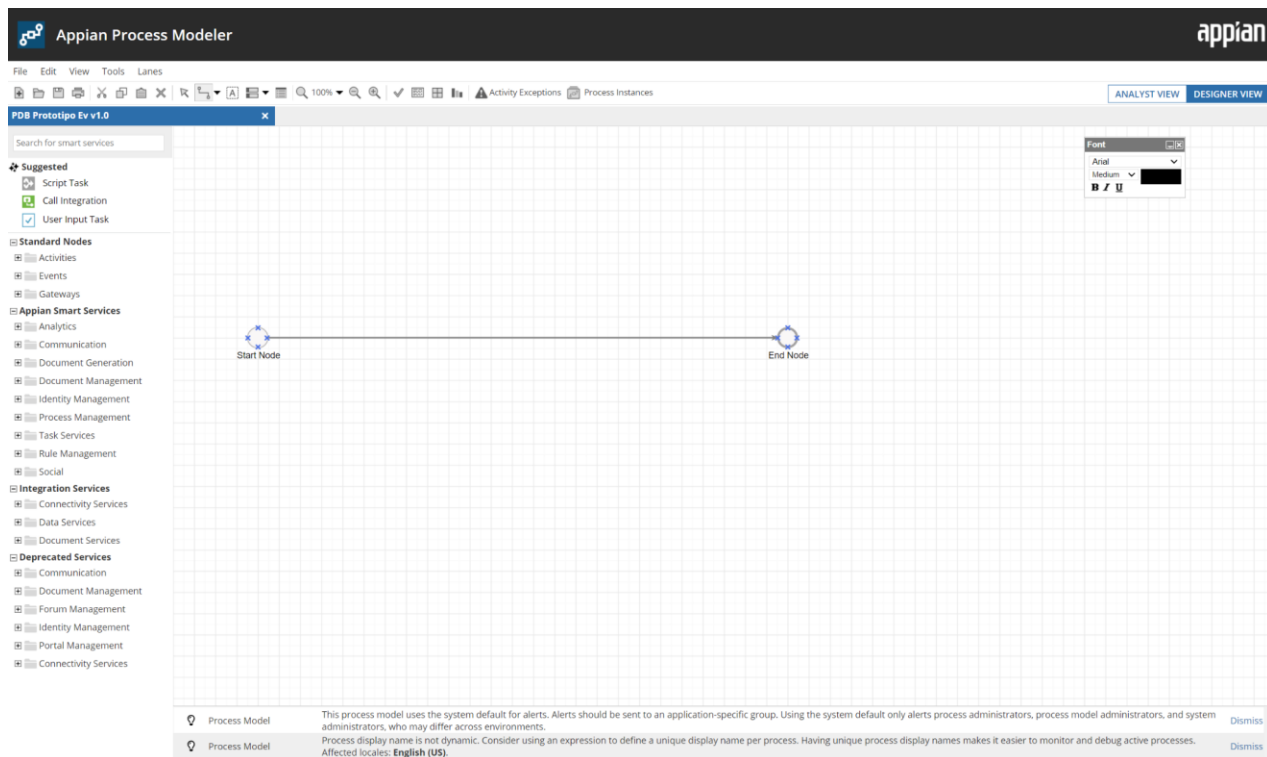
CREATE

Imagen: Creación de un Process Model.

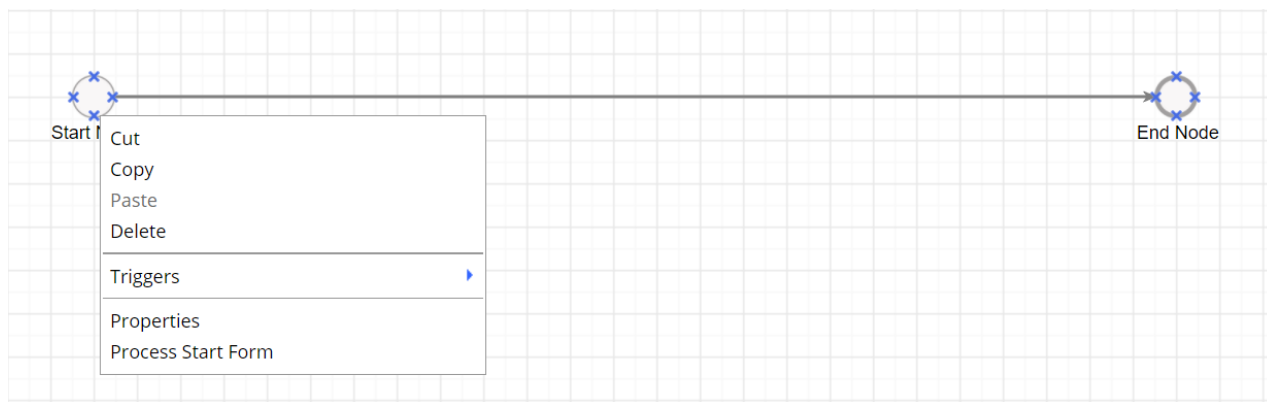
Appian Process Modeler

Al crear un **Process Model** se abre una ventana de Appian Process Modeler, donde puedes diseñar el Process Model.

Se comienza con dos nodos uno de inicio y otro que indica el final del proceso y una línea que los une definiendo la dirección del proceso.



Al seleccionar el nodo de inicio seleccionamos la opción de **Process Start Form** para diseñar el proceso que seguirá la aplicación que desarrollas.



A continuación seleccionamos la interfaz a la que queremos conectar la base de datos.

Process Model Properties » PDB Eva ✕

General Variables Process Start Form Deadlines Alerts Data Management

Edit Form

English (US)

Spanish (MX)

French (CA)

English (GB)

Chinese (CN)

Chinese (HK)

Dutch

French (FR)

German

Italian

Japanese

Polish

Portuguese

Russian

Spanish

Swedish

Arabic

Greek

Korean

☒ Enable this language

Interface

"PDB_Evaluacion"

Clear

Y después se crearán los parámetros necesarios para el Process Model de esa interfaz.

Process Model Properties » PDB Prototipo Ev

General Variables **Process Start Form** Deadlines Alerts Data Management

Edit Form

English (US) Spanish (MX) French (CA) English (GB) Chinese (CN) Chinese (HK) Dutch
French (FR) German Spanish Swedish
Arabic Greek Korean

☒ Enable this language

Interface
"PDB_Evaluacion"

Create Process Parameters

Do you want to automatically create process parameters to match your interface's inputs?

NO YES

CANCEL OK

Podemos ver que se crearon las Rule Input que usamos en el formulario que estamos desarrollando, **cancel** y **evaluacion**.

Process Model Properties » PDB Eva



General Variables **Process Start Form** Deadlines Alerts Data Management

Edit Form

English (US) Spanish (MX) French (CA) English (GB) Chinese (CN) Chinese (HK) Dutch
French (FR) German Italian Japanese Polish Portuguese Russian Spanish
Swedish Arabic Greek Korean

☒ Enable this language


Interface

"PDB_Evaluacion"



Clear

EDIT INTERFACE

Specify the data to pass to the interface  Refresh

Rule Input	Type	Multiple?	Value
cancel	Boolean	No	cancel
evaluacion	PDB_Evaluacion	No	evaluacion

CANCEL

OK

Ahora necesitamos crear una ruta para enviar el formulario si fue llenado y otra ruta para el botón de cancelar. Para esto ocuparemos un elemento **XOR** que se encuentra en **Gateways** del lado izquierdo de la pantalla. Observa que existen distintos elementos que puedes agregar a tu proceso.

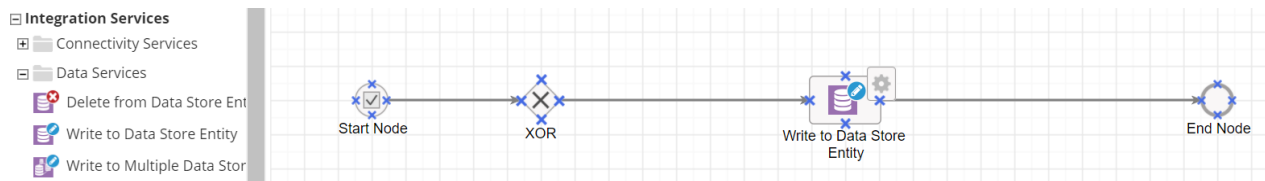
Standard Nodes

- Activities
- Events
- Gateways

- AND
- Complex
- OR
- XOR



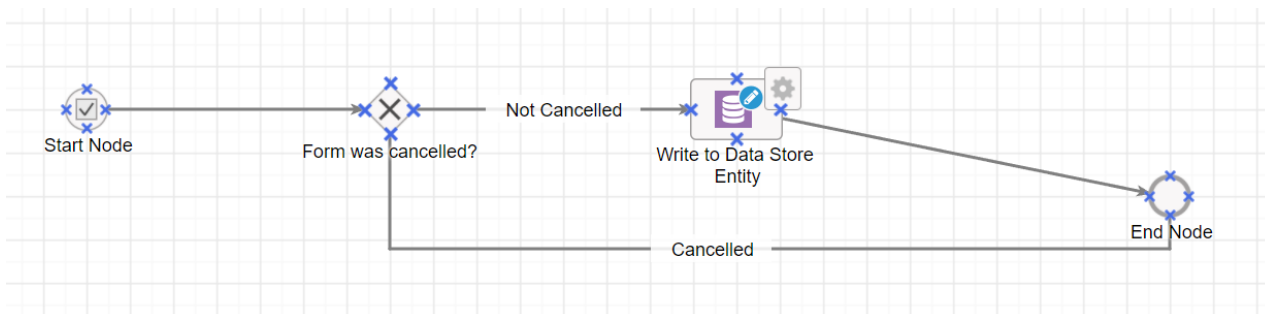
Agregamos un elemento **Write to Data Store Entity** en el proceso. Este elemento se encuentra dentro de **Data Services** que a su vez se encuentra dentro de **Integration Services**. Se utiliza para indicar el **Data Store** donde se guardarán los datos.



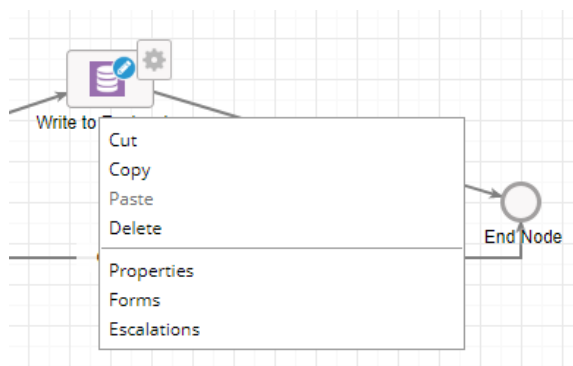
Ya tenemos los elementos necesarios para el proceso. Ahora necesitamos indicar los caminos que puede tomar el proceso. Agregaremos una flecha desde el elemento XOR hasta el nodo final. En el menú de Appian Process Modeler se encuentran las líneas que indican los caminos del proceso y otras opciones para realizar en el diseño del proceso.



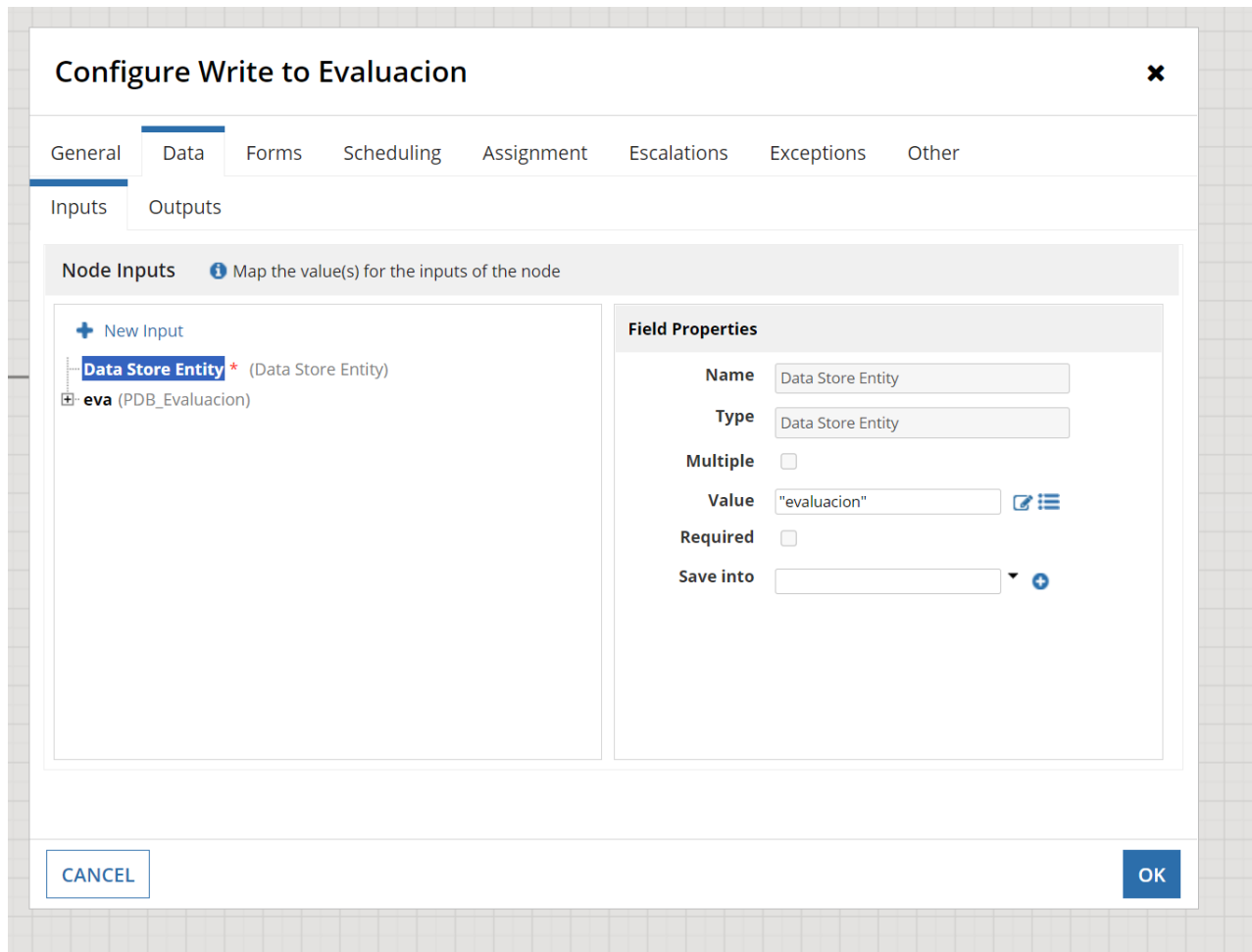
Observa que al iniciar el proceso nos dirigimos a un elemento de decisión, donde se pregunta si el formulario fue cancelado, si fue cancelado el proceso se dirige al nodo final. Por otro lado si se envió el formulario la información recabada se escribirá en la base de datos seleccionada. y después se terminará el proceso.



Ahora, debemos de configurar el elemento Write to Data Store Entity. Con clic derecho sobre el elemento ingresamos a **Properties** y en seguida a **Data**.



Seleccionamos **Data Store Entity** del lado izquierdo y en **Field Properties** seleccionamos la tabla dentro de nuestra base de datos en la opción de **Value**.



Configure Write to Evaluacion [Close]

General | **Data** | Forms | Scheduling | Assignment | Escalations | Exceptions | Other

Inputs | Outputs

Node Inputs ⓘ Map the value(s) for the inputs of the node

+ New Input

- Data Store Entity** * (Data Store Entity)
- eva (PDB_Evaluacion)

Field Properties

Name Data Store Entity

Type Data Store Entity

Multiple ☐

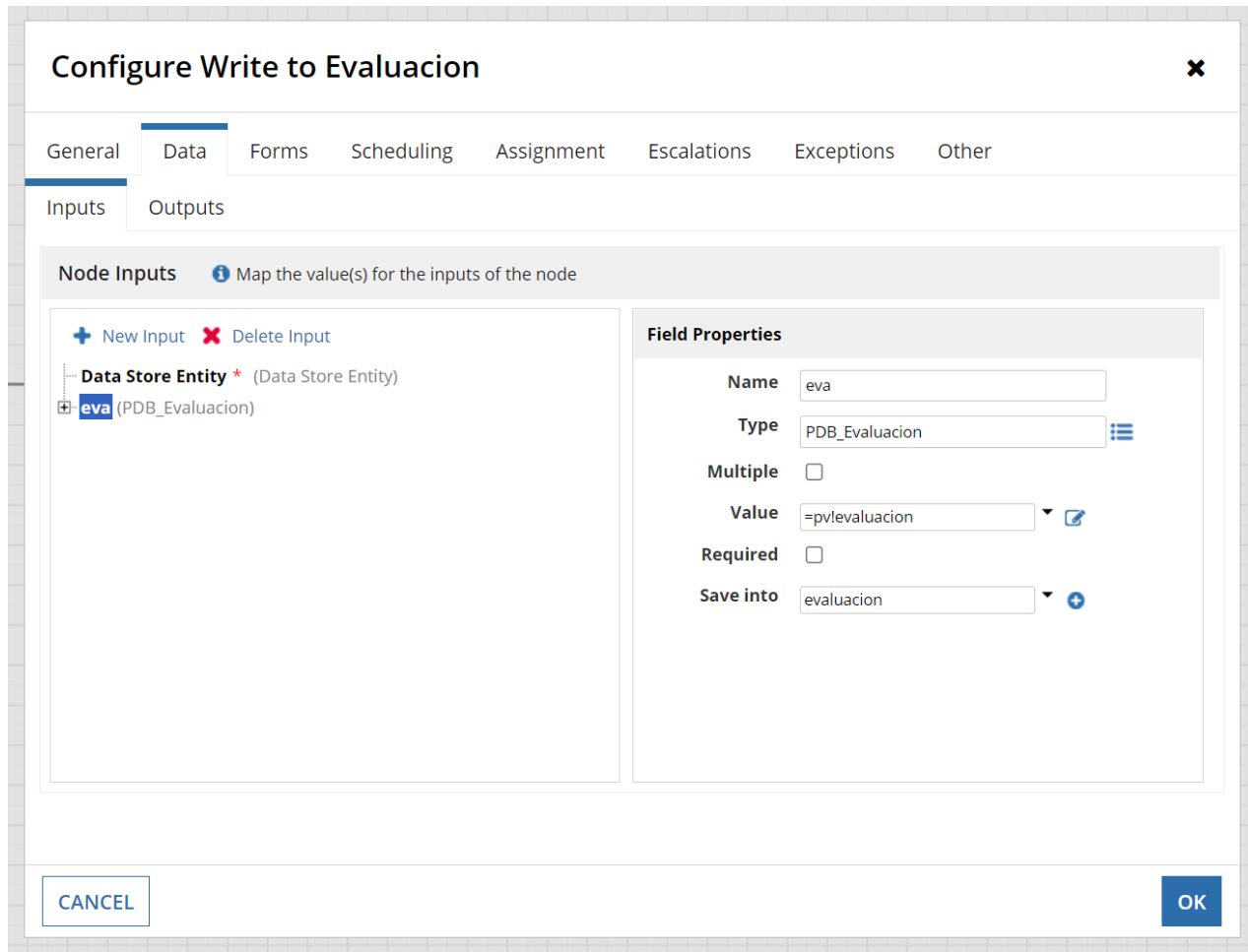
Value "evaluacion" [Link] [List]

Required ☐

Save into [Dropdown] [Add]

CANCEL OK

Después seleccionamos **New input**, le asignamos un nombre y en **Field Properties** seleccionamos nuestro Data Type en la opción de **Type**. Y de nuevo en **Value** seleccionamos la tabla de nuestra base de datos, de igual manera en **Save into**.



Configure Write to Evaluacion [X]

General | **Data** | Forms | Scheduling | Assignment | Escalations | Exceptions | Other

Inputs | Outputs

Node Inputs ⓘ Map the value(s) for the inputs of the node

+ New Input ✖ Delete Input

..... **Data Store Entity** * (Data Store Entity)

+ **eva** (PDB_Evaluacion)

Field Properties

Name eva

Type PDB_Evaluacion [Menu]

Multiple ☐

Value =pvlevaluacion [Dropdown] [Edit]

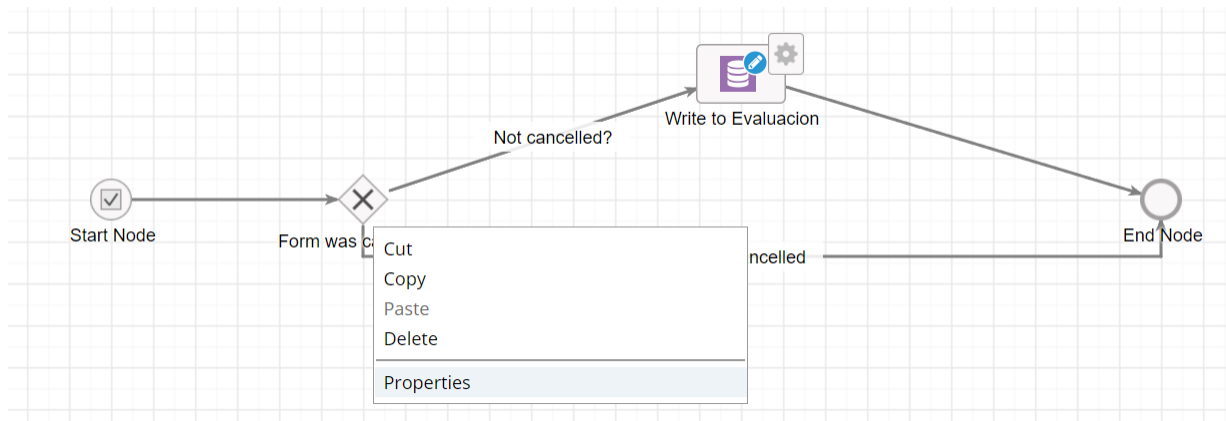
Required ☐

Save into evaluacion [Dropdown] [Add]

CANCEL OK

Con esto quedaría configurado el elemento Write to Data Store Entity.

Ahora tenemos que configurar el elemento **XOR**, con clic derecho sobre el elemento accedemos a **Properties** y en seguida a **Decision**.



Dentro del recuadro **Conditions** seleccionamos el parámetro **cancel**, así si el valor de **cancel** es true el proceso se dirige al nodo final y por otro lado si **cancel** es false los datos se escriben en la base de datos.

Configure Form was cancelled?

General
Decision

Flows

➔ **Incoming Path(s):** 1 or more paths can enter an XOR gateway

➔ **Outgoing Paths:** 1 or more paths can exit an XOR gateway, but only ONE gets executed

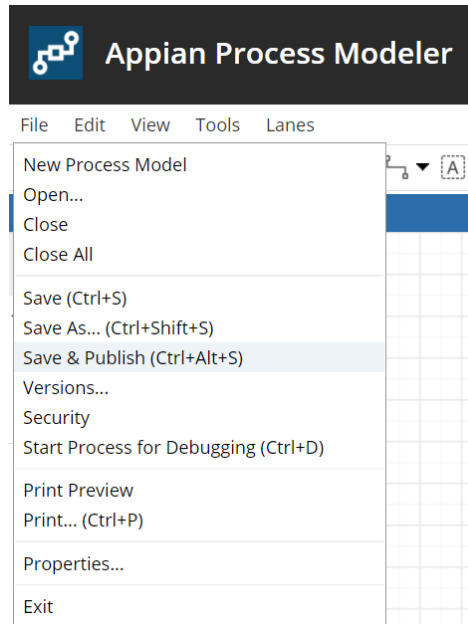
Conditions

	Condition	Result	Path Label	Order
✖	If <input type="text" value="pv\cancel"/> is True	go to <input type="text" value="End Node"/>	<input type="text" value="Cancelled"/>	
	Else if no rules are TRUE	go to <input type="text" value="Write to Evaluacion"/>		

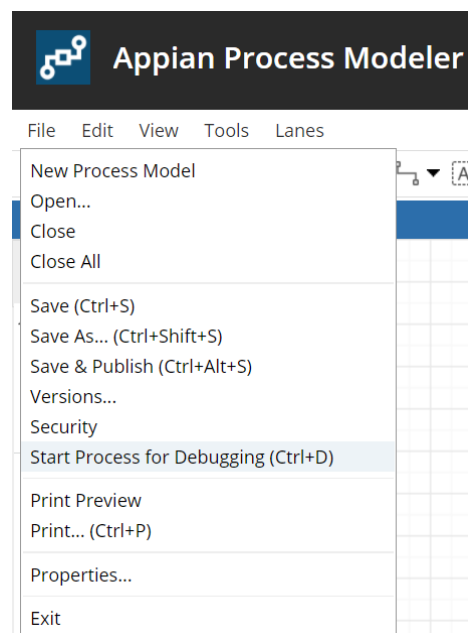
[NEW CONDITION](#)

[CANCEL](#)
[OK](#)

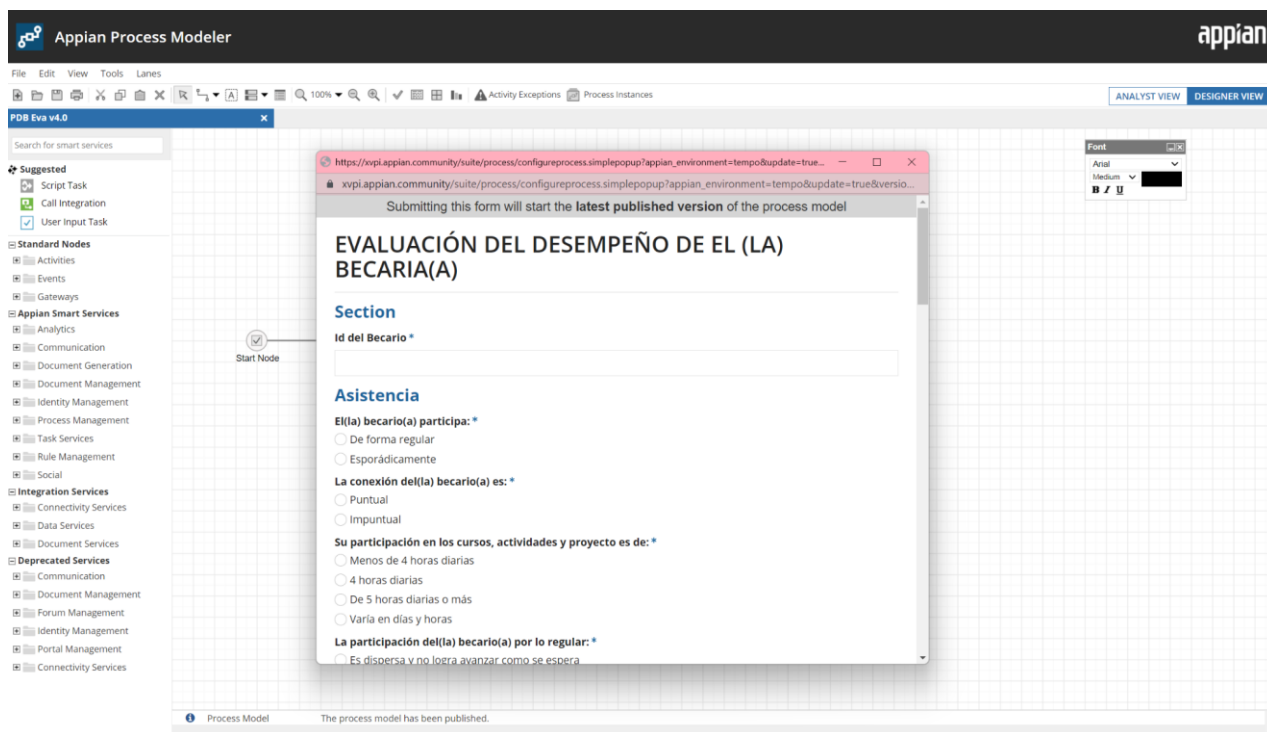
Con esto queda configurado el Process Model. Ahora solo hace falta guardar los cambios y publicar. Damos clic en **Save & Publish**, es muy importante no solo guardar los cambios sino publicarlos también.



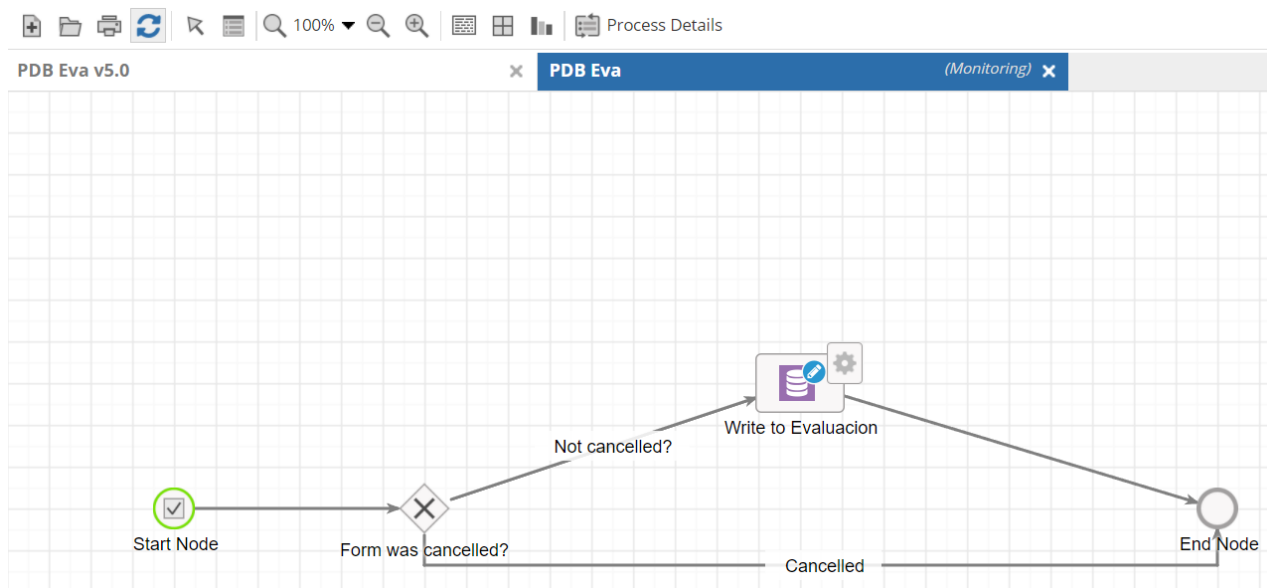
Para comprobar que el proceso está bien diseñado y funciona adecuadamente es conveniente hacer una prueba del proceso. Esto lo hacemos en la opción **Start Process for Debuggin**.



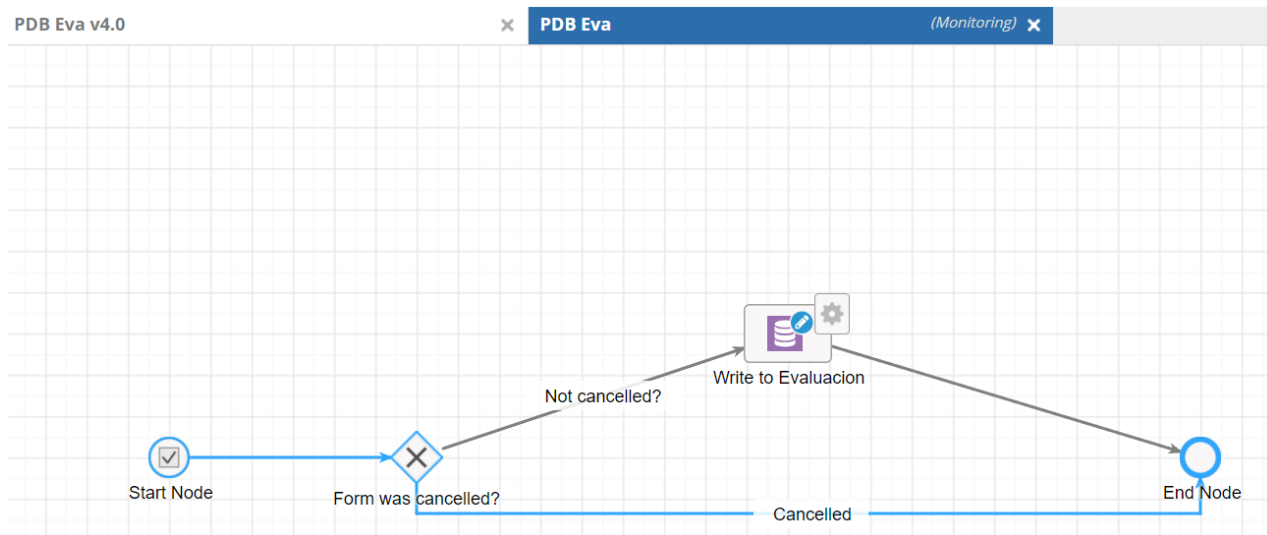
En esta opción podemos probar la interfaz que desarrollamos y verificar su funcionamiento. En la nueva ventana aparecerá la interfaz.



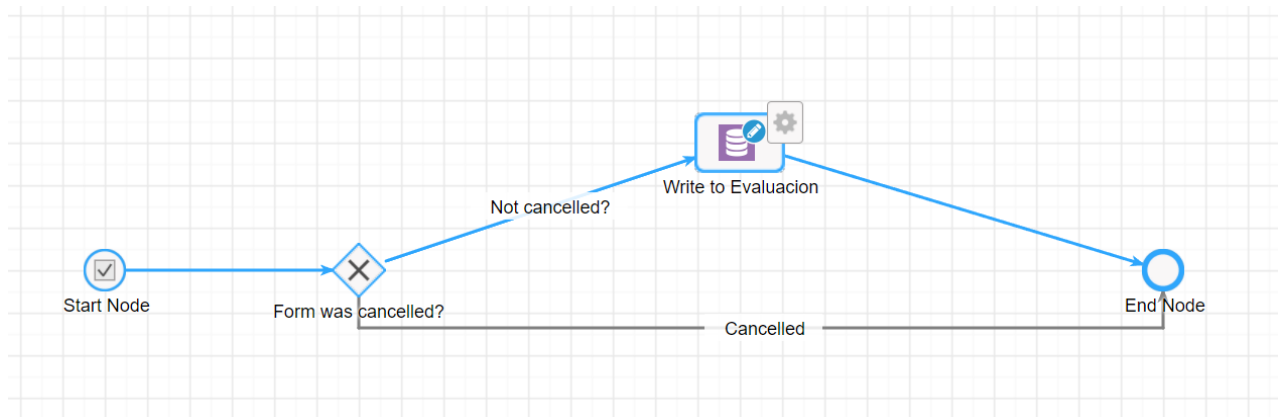
Si probamos cancelando el formulario nos mandará a una nueva sección llamada Monitoring donde deberíamos ver el diseño del proceso comenzado, como se muestra en la imagen de abajo.



Después debemos dar clic en el icono de **Refresh** (dos flechas azules en círculo) para que continúe el proceso. Si todo está configurado adecuadamente deberías de ver en azul el flujo del proceso al seleccionar el botón de cancelar de tu formulario.



Si volvemos a seleccionar **Start Process for Debuggin**, y ahora respondemos el formulario al enviarlo y siguiendo las indicaciones anteriores deberíamos de ver el flujo en azul de nuestra aplicación.



Ahora, si revisas la base de datos ya debería de mostrarse la información que se llenó en el formulario.

	idbecario	nombrecario	participa	conexion	participacioncursos	participacionregular	cargaacademica	actitud	error	plazos	reportes	interescapacitacion	interescontinuar	conocimiento	inve
	1		De forma regular	Puntual	Menos de 4 horas diarias	Se dirige a realizar mayoritariamente trabajos o actividades personales	Permanece más de 4 horas debido a que realiza trabajos personales	Es individualista pero realiza las actividades adecuadamente	Admite su falla y reinicié en lo mismo	Nunca	Confusa	Nunca	Para apoyar al proyecto al que fue asignado	Regulares	Nunc

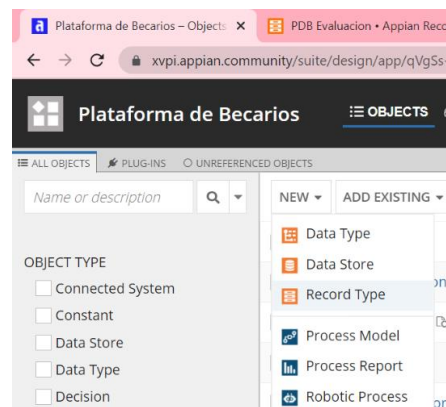
Con esto terminamos de crear completamente los procesos necesarios para que un formulario desarrollado en Appian funcione correctamente.

Creación de Reportes con gráficas en Appian.

Para crear un reporte graficando la información obtenida de una base de datos necesitamos crear una serie de elementos en la aplicación de Appian. A continuación ejemplificamos su creación.

Record Type

Primero debemos de crear un Record Type en la aplicación.



A continuación configuramos la tabla de donde queremos extraer la información.

Configure Data Source

Choose Database Table [?](#)

☒ Browse ☐ Search

jdbc/Appian (Tomcat)


- exvfmvehicle
- hrassessment
- hrcandidatedetails
- pdbevaluacion** ✓
- Person
- Product

Preview

ldbecario	nombrebecario	participa	conexion	participacioncursos	participacionregular	cargaacademica
Number (Integer)	Text	Text	Text	Text	Text	Text
1	-	De forma regular	Puntual	Menos de 4 horas diarias	Se dirige a realizar mayoritariamente tra...	Permanece más de 4 horas
2	-	Esporádicamente	Puntual	4 horas diarias	Es dispersa y no logra avanzar como se e...	Permanece más de 4 horas

[GO BACK](#) [CANCEL](#) [NEXT](#)

Después de seleccionar la tabla se mostrará en el Record Type los atributos que contiene la tabla. Guardamos los cambios.


PDB Evaluacion

DATA

- Data Model**
- Default Filters
- Data Sync

USER INTERFACES

- Tempo
- List
- Search and User Filters
- Views and Header
- Record Actions
























































MONITORING

- Performance

Data Model

Data Structure
Data Preview

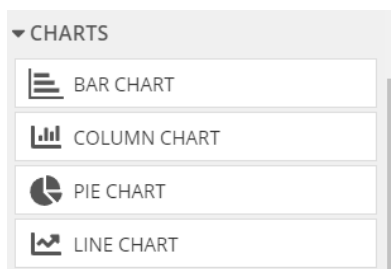
CONFIGURE FIELDS
NEW CUSTOM RECORD FIELD

Field Name	Type			
idbecario 	Number (Integer)			
nombrecario	Text			
participa	Text			
conexion	Text			
participacioncursos	Text			
participacionregular	Text			
cargaacademica	Text			
actitud	Text			
error	Text			
plazos	Text			
reportes	Text			
interescapacitacion	Text			
interescontinuar	Text			
conocimiento	Text			
investigar	Text			
propositivo	Text			
calificacion	Number (Integer)			
conocimientosextras	Text			

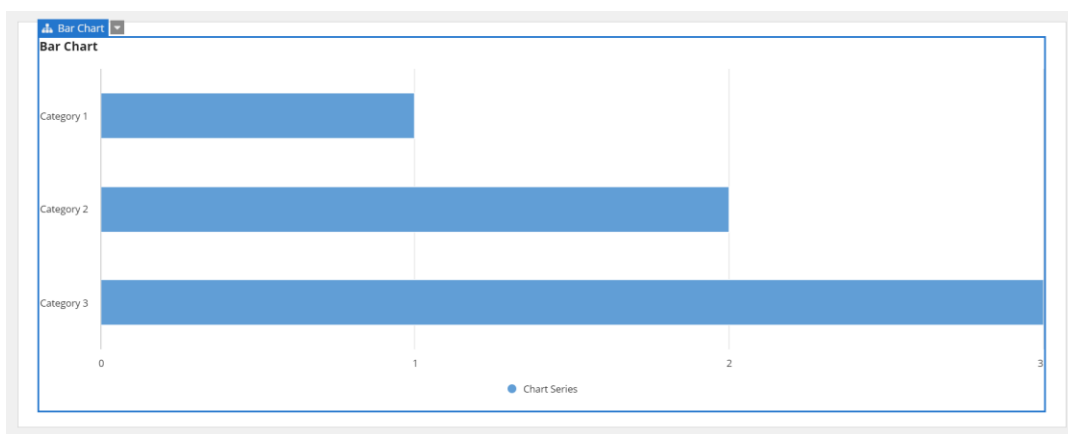
Después de crear el Record Type podemos crear la interfaz donde se mostrará la información recabada.

Como hicimos anteriormente, creamos una interfaz y accedemos a Design Mode.

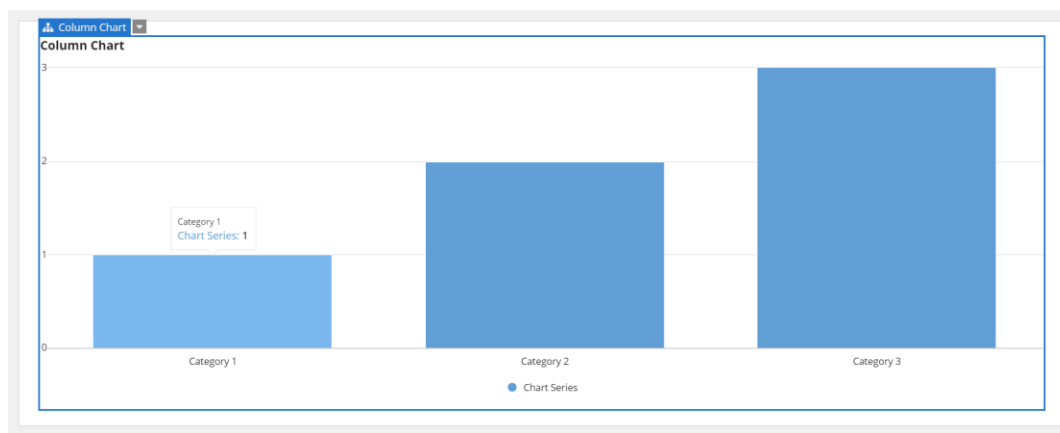
Design Mode cuenta con varios componentes drag and drop para diseñar una interfaz, en esta ocasión haremos uso de los componentes de tipo Charts.



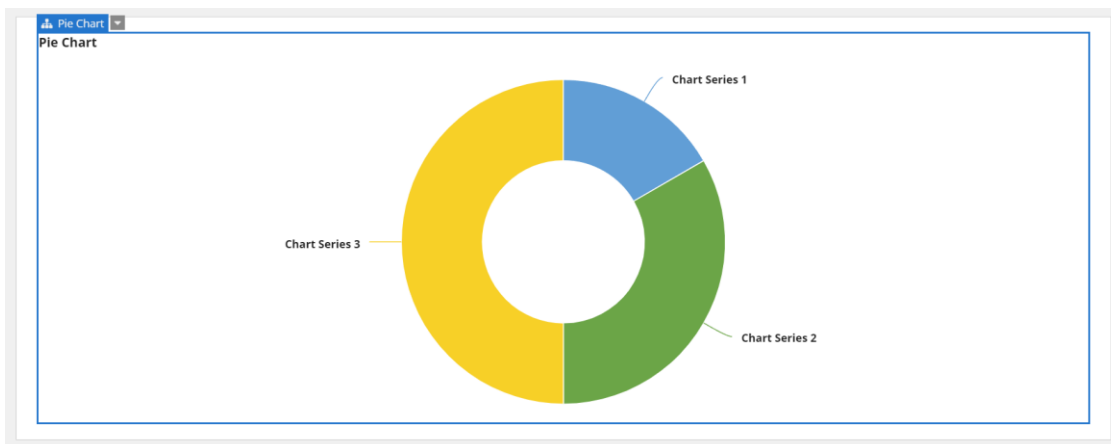
Ejemplo del componente Bar Chart:



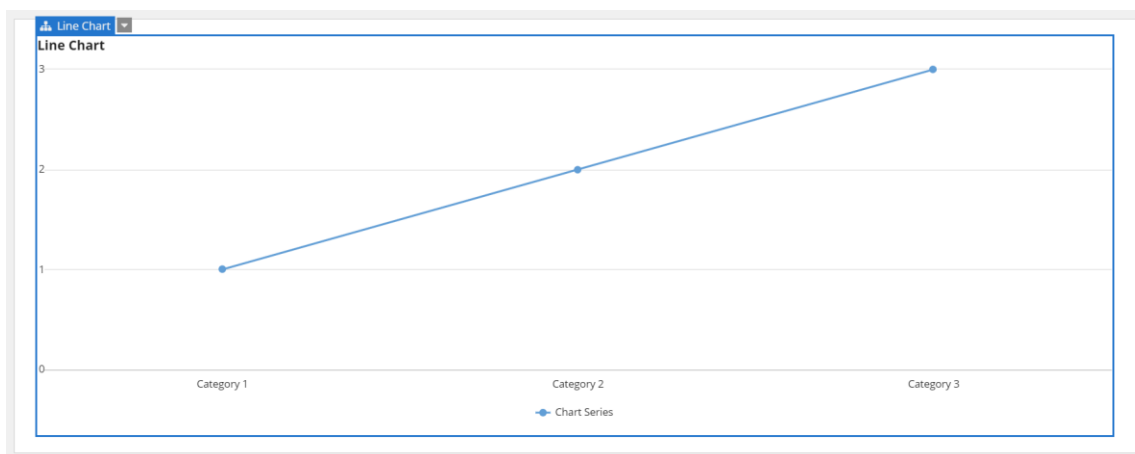
Ejemplo del componente Column Chart:



Ejemplo del componente Pie Chart:



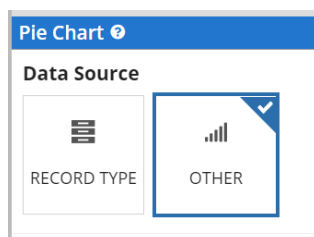
Ejemplo del componente Line Chart:



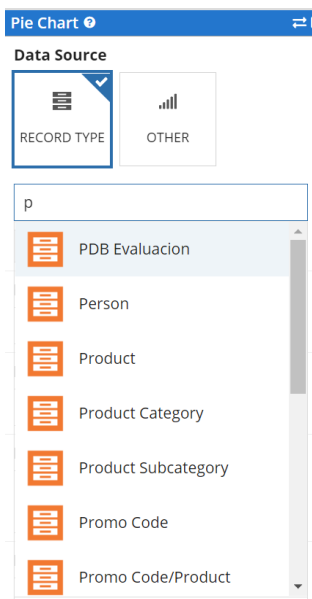
Como hemos visto, al seleccionar un componente del lado derecho podemos editar sus configuraciones.

Al seleccionar alguno de los componentes puedes hacer las configuraciones siguientes. En nuestro caso utilizaremos componentes Pie Chart.

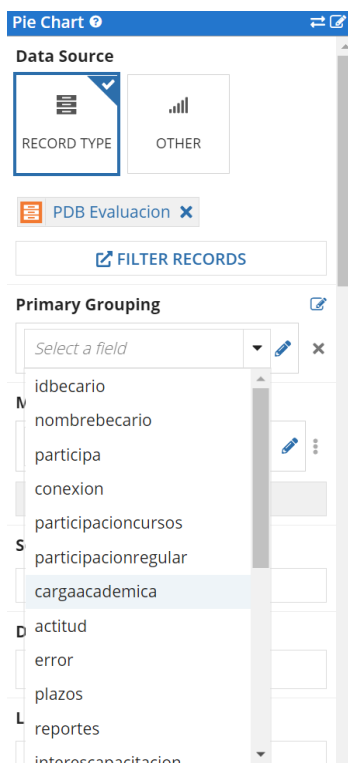
Seleccionamos un componente Pie Chart, la configuración más importante en este tipo de componentes es elegir el **Record Type** de donde se obtendrá la información para graficar.



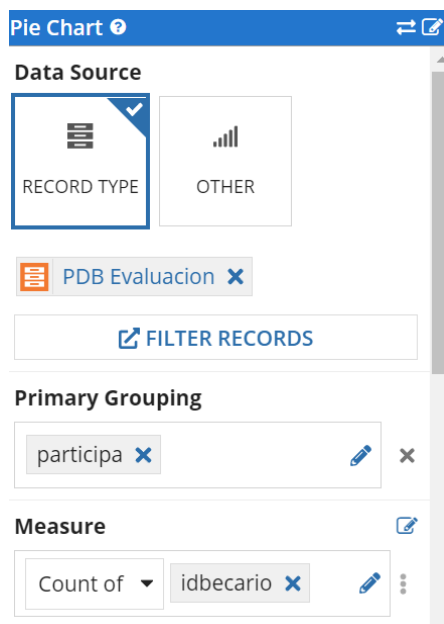
Seleccionamos el Record Type que creamos de la tabla de nuestra base de datos.



Y a continuación seleccionamos el atributo de la tabla del cual queremos graficar los datos recabados en la opción de **Primary Grouping**.



Y en la opción de **Measure** podemos agregar un atributo con el que se contarán los datos, por default se selecciona la llave primaria de tu tabla. Nosotros la dejaremos así.



Pie Chart

Data Source

RECORD TYPE (selected) OTHER

PDB Evaluacion

FILTER RECORDS

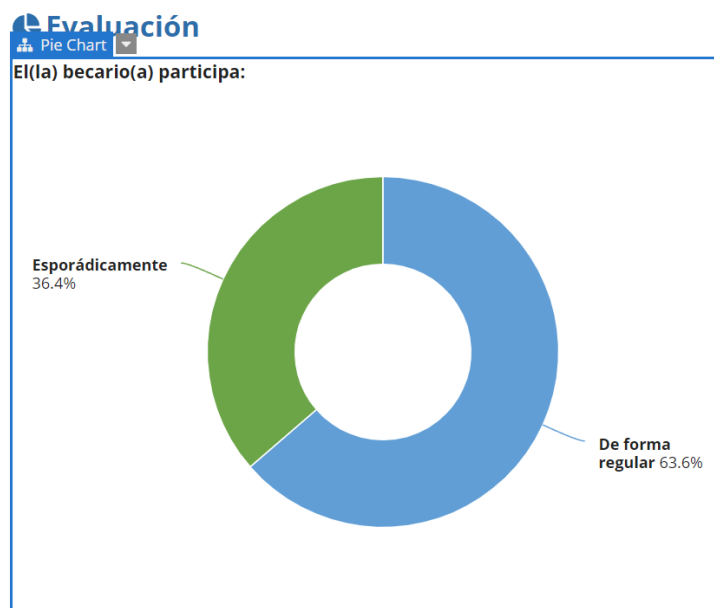
Primary Grouping

participa

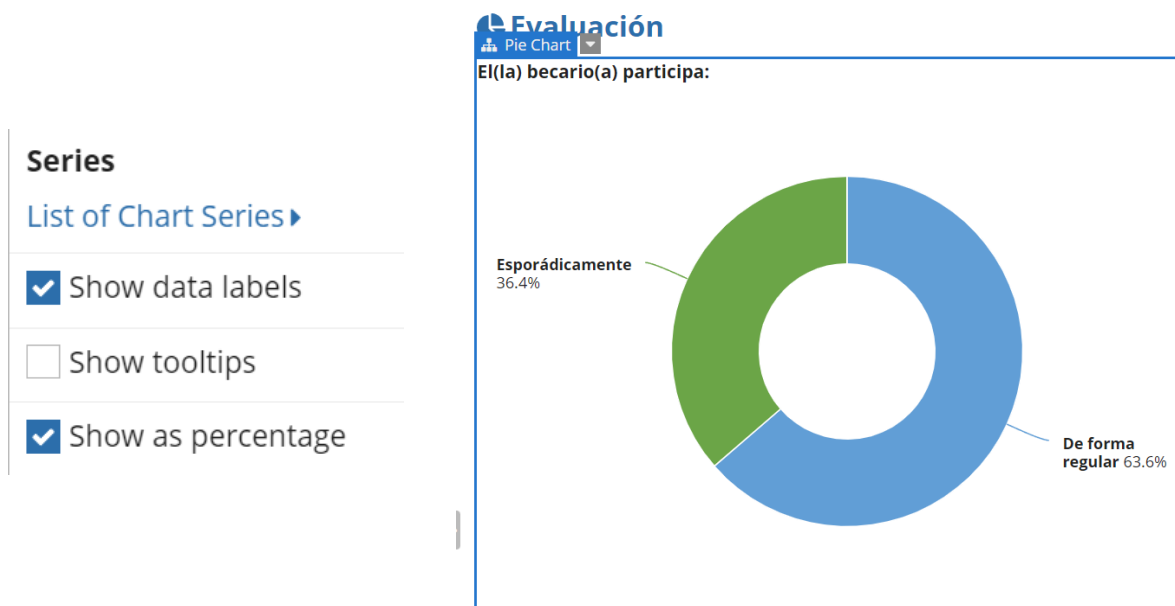
Measure

Count of idbecario

Esto es suficiente para mostrar la información en la gráfica.




En la opción **Series** puedes configurar si se mostrará la cantidad de elementos que pertenecen a un grupo y de qué forma se mostrará, por elemento o en porcentaje.



Además es posible modificar los colores mostrados en la gráfica

Classic ▼



[Color Best Practices](#)

Style

Donut ▼

Y en la opción Style es posible mostrar la gráfica en forma de pastel o de dona como se muestra en las imágenes de abajo.

Style

Pie ▼

Series Labels

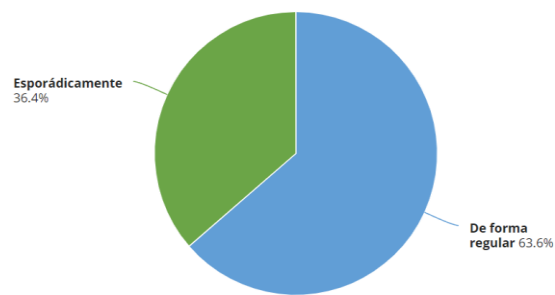
In legend ▼

Height

Short ▼

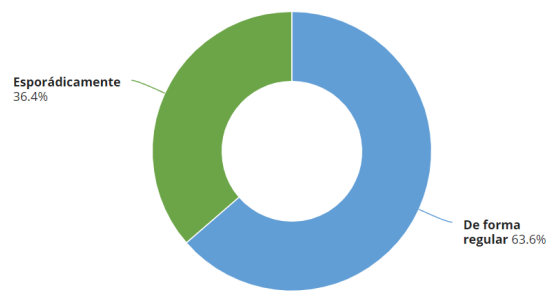
Evaluación

El(la) becario(a) participa:

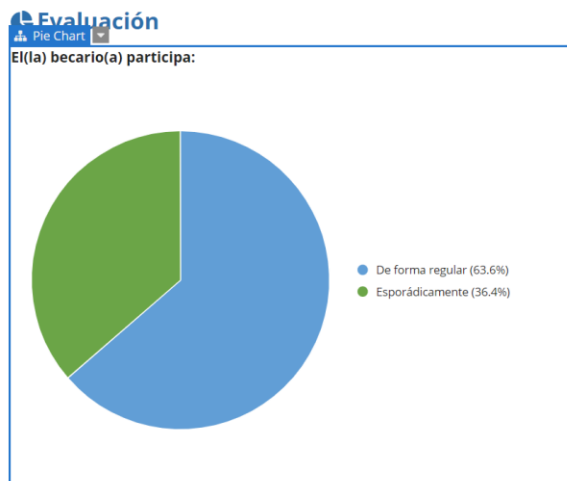


Evaluación

El(la) becario(a) participa:



En la opción Height es posible modificar el tamaño de la gráfica. Y por último en la opción Series Labels puedes mostrar los datos a un lado de la gráfica o señalando cada grupo.



Esto es suficiente para hacer una interfaz que muestre los datos recabados en un formulario de forma gráfica.





CONECTAR UNA BASE DE DATOS POSTGRESQL


En el siguiente link encontraremos información para conectar algunas bases de datos relacionales.

https://docs.appian.com/suite/help/21.1/Configuring_Relational_Databases.html

Algunos drivers disponibles para conectarse son los siguientes:

RDBMS	Driver
MySQL 8.0 with InnoDB engine	Connector/J 5.1.x
MySQL 5.7 with InnoDB engine	Connector/J 5.1.x
MariaDB Server 10.5	MariaDB Connector/J 2.6 Series
MariaDB Server 10.4	MariaDB Connector/J 2.6 Series
Oracle 19c	Oracle 19.7 JDBC Driver Certified with JDK8
Oracle 18c	Oracle 19.7 JDBC Driver Certified with JDK8
Oracle 12c R2	Oracle 19.7 JDBC Driver Certified with JDK8
Oracle 12c R1	Oracle 19.7 JDBC Driver Certified with JDK8
Oracle 11g R2	Oracle 19.7 JDBC Driver Certified with JDK8
Microsoft SQL Server 2019	Microsoft JDBC SQL Server Driver 8.x
Microsoft SQL Server 2017	Microsoft JDBC SQL Server Driver 8.x
Microsoft SQL Server 2016	Microsoft JDBC SQL Server Driver 8.x
Microsoft SQL Server 2014	Microsoft JDBC SQL Server Driver 8.x
Microsoft SQL Server 2012	Microsoft JDBC SQL Server Driver 8.x
Microsoft SQL Server 2008 (Deprecated)	Microsoft JDBC SQL Server Driver 2.x and 7.x
PostgreSQL 12	PostgreSQL JDBC 4.2 Driver
PostgreSQL 11	PostgreSQL JDBC 4.2 Driver

Para conectar una base de datos externa a Appian nos dirigimos a **Appian Administration Console**. En el apartado de **Integration** seleccionamos **Data Sources**. Y a continuación New Data Source.



Maintenance Window

SAML

Users

Web API Authentication

DEVOPS

Deployment

Health Check

Infrastructure

INTEGRATION

Certificates

Data Sources >

Document Extraction

Email

EXPORTIMPORT

appian

Data Source Management

New way to connect data sources. If you want to deploy data sources to other environments and restrict who can access data sources during development, you can create connected systems for data sources instead. [Learn more about data source connected systems](#)

Name	Type	Connection String	Last Modified
No items available			
+ New Data Source			

Configure Data Source

Name *

Example: jdbc/ExampleName

Username *

Password *

Type *

Select Type

Connection String *

Example: jdbc:

CANCEL

TEST CONNECTION


SAVE

Establecemos el nombre, el username de Postgres, la contraseña, el tipo y una cadena para conectarse con postgres, de la cual proporciona un ejemplo de cómo crearla.

Configure Data Source

Name *	Username *
<input type="text" value="jdbc/PostgreSQL"/>	<input type="text" value="postgres"/>
Example: jdbc/ExampleName	
Type *	Password *
<input type="text" value="PostgreSQL"/>	<input type="password" value="....."/>
Connection String *	
<input type="text" value="jdbc:postgresql://localhost:5432/comedor"/>	
Example: jdbc:postgresql://yourserver.example.com:5432/your_db_name	

Esto es suficiente para conectar una base de datos externa a Appian. Ahora al crear un **Data Store** en el caso de postgres debes seleccionar jdbc/PostgreSQL en la opción de Data Source o en otro caso seleccionar la fuente de datos que hayas agregado.

 **Appian Data Store Designer**

comedor
Enter data store description here

Data Source

Data Entities

RESULTADOS

Se investigó la herramienta de low code Appian, sus requerimientos de uso y precios. Se exploró y se hizo una prueba de desarrollo en su plataforma de low code. Se encontró documentación y cursos que enseñan cómo usar esta herramienta.

- Se logró crear un formulario con la herramienta Appian.
- Se logró guardar los datos recabados del formulario en una base de datos a través de Appian.
- Se logró conectar la información de la base de datos con una interfaz para graficar los resultados con el uso de Appian.

CONCLUSIONES:

Después de explorar la herramienta de low code de Appian concluí que su uso está dirigido a personas que saben desarrollar software, ya que a pesar de tener una plataforma de drag and drop, es necesario contar con conocimientos técnicos los para la construcción de aplicaciones, como el diseño, creación y uso de las bases de datos, entre otros.

Appian cuenta con una gran documentación y tutoriales para aprender a usar su plataforma de low code, que es adecuado para un equipo que desarrollará en Appian distintos proyectos o una aplicación que se usará por largo tiempo. Sin embargo es necesario revisar bastante material para desarrollar una aplicación sencilla y no es conveniente si no se va a seguir haciendo uso de la herramienta.

BIBLIOGRAFÍA/REFERENCIAS

¿Qué es una plataforma low-code? | Velneo. (s/f). Recuperado el 30 de noviembre de 2021, de <https://velneo.es/que-es-plataforma-lowcode/>

El árbol de decisión para las categorías Low-Code. (s/f). AuraQuantic. Recuperado el 30 de noviembre de 2021, de <https://www.auraquantic.com/es/que-es-low-code/>

Las mejores plataformas de Low-Code. (2021, enero 13). DataScope. <https://mydatascope.com/blog/es/las-mejores-plataformas-de-low-code/>

Appian: Automatización Low-Code, Aplicaciones Empresariales, BPM, RPA. (s/f). Recuperado el 3 de diciembre de 2021, de <https://appian.com/es.html>

Vincent, A. P., Natis, Y., Iijima, K., Wong, J., Ray, S., & Jain, A. (s/f). Magic Quadrant for Enterprise Low-Code Application Platforms. 32

Appian. (2019, mayo 29). *Become an Appian Expert Fast in this Intro to Appian*. https://www.youtube.com/watch?v=Sk6HuQ_klxY

Augustine C. (2020, marzo 1). *APPIAN TUTORIAL: BUILDING A SIMPLE TIMESHEET APPLICATION*. <https://www.youtube.com/watch?v=aJXle5ftPM8>

Feedback. (s/f). Google Docs. Recuperado el 21 de enero de 2022, de https://docs.google.com/forms/d/18f1u2V13q1Ef1mPqFNFhX9BHwd1rdq7S-H9TAv1XLAw/viewform?usp=embed_facebook

Low-code and shadow IT | Genus Blog. (s/f). Genus. Recuperado el 21 de enero de 2022, de <https://www.genus.no/blog/low-code-and-shadow-it/>

5 Reasons to Use Low-Code Platforms in Higher Education. (s/f). Recuperado el 21 de enero de 2022, de <https://www.bplogix.com/blog/5-reasons-to-use-low-code-platforms-in-higher-education>

How can low-code help educational institutions? (2020, diciembre 23). Decode - A Publication by Zoho Creator. <https://www.zoho.com/creator/decode/how-can-low-code-help-educational-institutions>

July 27, J. G. in D. on, 2020, & Pst, 8:20 Am. (s/f). *Universities turning to low code to help bring back students amid COVID-19.* TechRepublic. Recuperado el 21 de enero de 2022, de <https://www.techrepublic.com/article/universities-turning-to-low-code-to-help-bring-back-students-amid-covid-19/>

Blog. (s/f). NoCodeDev. Recuperado el 21 de enero de 2022, de <https://nocode.dev/blog/>

KibiOne. (2021, enero 12). *7 Best No-Code App Builders 2021 (+ What You Can Build).* https://www.youtube.com/watch?v=3_ftZdc-sYI

Review these 9 low-code use cases and industry examples. (s/f). SearchSoftwareQuality. Recuperado el 21 de enero de 2022, de <https://searchsoftwarequality.techtarget.com/tip/Review-these-9-low-code-use-cases-and-industry-examples>

OutSystems Announces Increased Adoption of Low-Code by Universities to Accelerate Digital Innovation. (2019, marzo 28). <https://www.businesswire.com/news/home/20190328005109/en/OutSystems-Announces-Increased-Adoption-of-Low-Code-by-Universities-to-Accelerate-Digital-Innovation>

Coaching No Code Apps. (2020, octubre 21). *How to Choose the Right No Code App Development Platform.* <https://www.youtube.com/watch?v=zGeEa0XFhAA>

El low-code no es nada nuevo. (s/f). Recuperado el 21 de enero de 2022, de <https://disfrutaprogramando.com/el-low-code-no-es-nada-nuevo/>

Qué dicen del low-code las grandes consultoras. (s/f). Recuperado el 21 de enero de 2022, de <https://disfrutaprogramando.com/que-dicen-del-low-code-las-grandes-consultoras/>

Pros y contras que debes tomar en cuenta al pensar en low-code. (s/f). Recuperado el 21 de enero de 2022, de <https://thestandardcio.com/2020/08/25/low-code-pros-y-contras/>

Low-code, ¿un riesgo para la seguridad TI? (s/f). Recuperado el 21 de enero de 2022, de <https://www.muycanal.com/2021/04/28/low-code-seguridad-ti>

admin. (s/f). *No-Code / Low-Code: Por qué debe tener cuidado.* Recuperado el 21 de enero de 2022, de <https://hitechglitz.com/spanish/no-code-low-code-por-que-debe-tener-cuidado/>

Manuel Zapata. (2021, febrero 27). *Plataformas No-Code y Low-Code | Charlando con los Null*Pointers.* <https://www.youtube.com/watch?v=78yGifQyukY>

CURSOS DE APPIAN CONSULTADOS PARA EL PROTOTIPO:

Appian Community. (s/f). Recuperado el 21 de enero de 2022, de <https://community.appian.com/>

- Interfaces 101: Build Forms and Other Interfaces
- Tutorial: Build a Low-Code App (~30 mins)
- Tutorial: Building a Form
- Tutorial: Building a Process Model
- Manage Data Easily with Custom Data Types
- Tutorial: Building a Record
- Reports: Build Basic Charts and Grids

GLOSARIO

workflow: Flujo de trabajo.

low-code: Desarrollo de aplicaciones con poco código.

no-code: Desarrollo de aplicaciones sin código.

TI: Tecnología de la información.

TIC: Tecnología de la Información y Comunicación.

IDE: Entorno de desarrollo integrado.

interfaz de usuario: Espacio donde se producen las interacciones entre seres humanos y máquinas.

back-ends: Área lógica de una aplicación.

servicio: Servicios de computación que ofrecen aplicaciones.

depurar: Quitar de una cosa lo malo, lo que es extraño o lo que no sirve para dejarla pura.

caso de uso: Descripción de una acción o actividad.

from scratch: Desde el inicio.

base de datos: Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.

llave primaria: Campo o combinación de campos que identifica de forma única a cada fila de una tabla. No puede haber dos filas en una tabla que tengan la misma llave primaria.



CRÉDITOS

Wendy Hernández Méndez

Diseño de la prueba de concepto: Luz María Ramírez Romero

Revisión: Luz María Ramírez Romero y Maribel Carmona Herrera