



PROGRAMACION DE APLICACIONES MÓVILES

PGY4121

Profesor:

DuocUC



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES





Actividad N°1: Iniciando un proyecto

El mundo híbrido de las App's

Objetivos

Lo que se espera que aprendas en esta Actividad es:

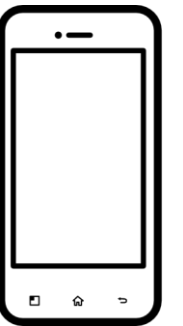
- » Conocer el concepto híbrido y por que se aplica a desarrollo de aplicaciones móviles
- » Identificar los distintos framework en el desarrollo de aplicaciones híbridas



**El mundo de las
app's**

PGY4121 | Las aplicaciones

- » En la actualidad, uno de los elementos tecnológicos más usados por gran parte de la población es el SmartPhone.
- » Usamos el celular para comunicarnos, informarnos, trabajar, realizar compras, escuchar música, ver series, ver nuestros datos bancarios, transferir, comprar online, etc.



PGY4121 | Las aplicaciones

- » ¿Hace cuanto fue que revisaste tu celular por última vez?
- » ¿Dónde dejas el celular al momento de dormir?



PGY4121 | Las aplicaciones

- » ¿Qué aplicaciones usan regularmente?
- » ¿Cuál es la aplicación que más usan?
- » ¿Qué no les gusta de las aplicaciones?



PGY4121 | Las aplicaciones

- » Ya forman parte de nuestras vidas cotidianas y sería difícil pensar en un mundo sin ellas!!



- » Hasta ahora hemos analizado el contexto del desarrollo de las aplicaciones móviles.
- » Ahora es la hora de entender que es un desarrollo híbrido y que herramientas tenemos para comenzar nuestro proyecto!

COMENCEMOS!



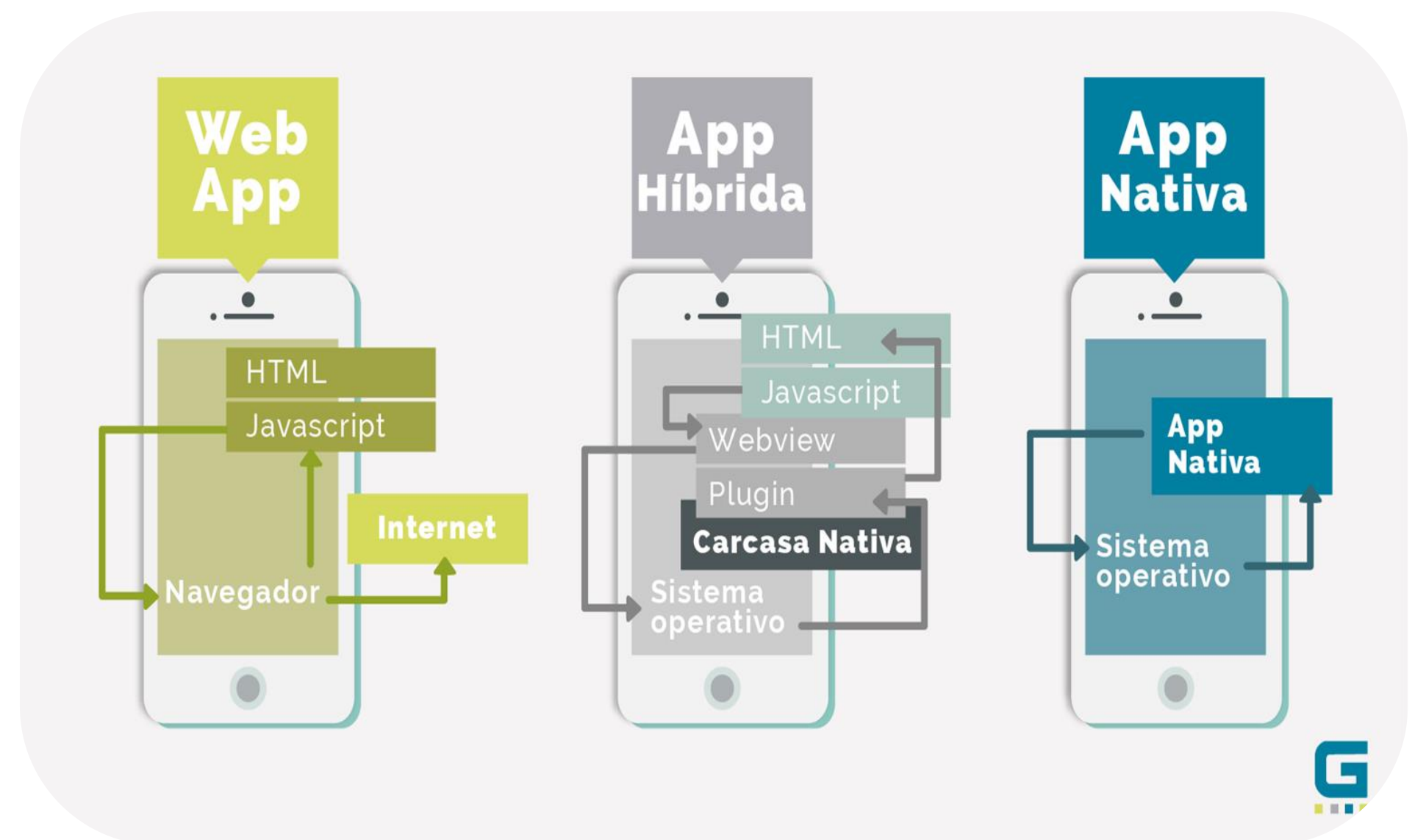
Desarrollo Híbrido y Ionic

- » El desarrollo Híbrido nos permite realizar una aplicación en un solo lenguaje apoyados en un Framework, que posteriormente nos servirá como base para distribuirlo en distintas plataformas tales como Android o IOS.



PGY4121 | Desarrollo Híbrido

- » El desarrollo Híbrido nos permite realizar una aplicación en un solo lenguaje apoyados en un Framework, que posteriormente nos servirá como base para distribuirlo en distintas plataformas tales como Android o IOS.



PGY4121 | ¿Por qué Híbrido?

- » Se codifica una vez y se despliega en cualquier lugar (Multiplataforma)
- » Usa el talento que ya se tiene (Comunidad Web)
- » Entrega un gran UX (User Experience) por medio de multi plataformas
- » Construye para el futuro mediante desarrollo Web

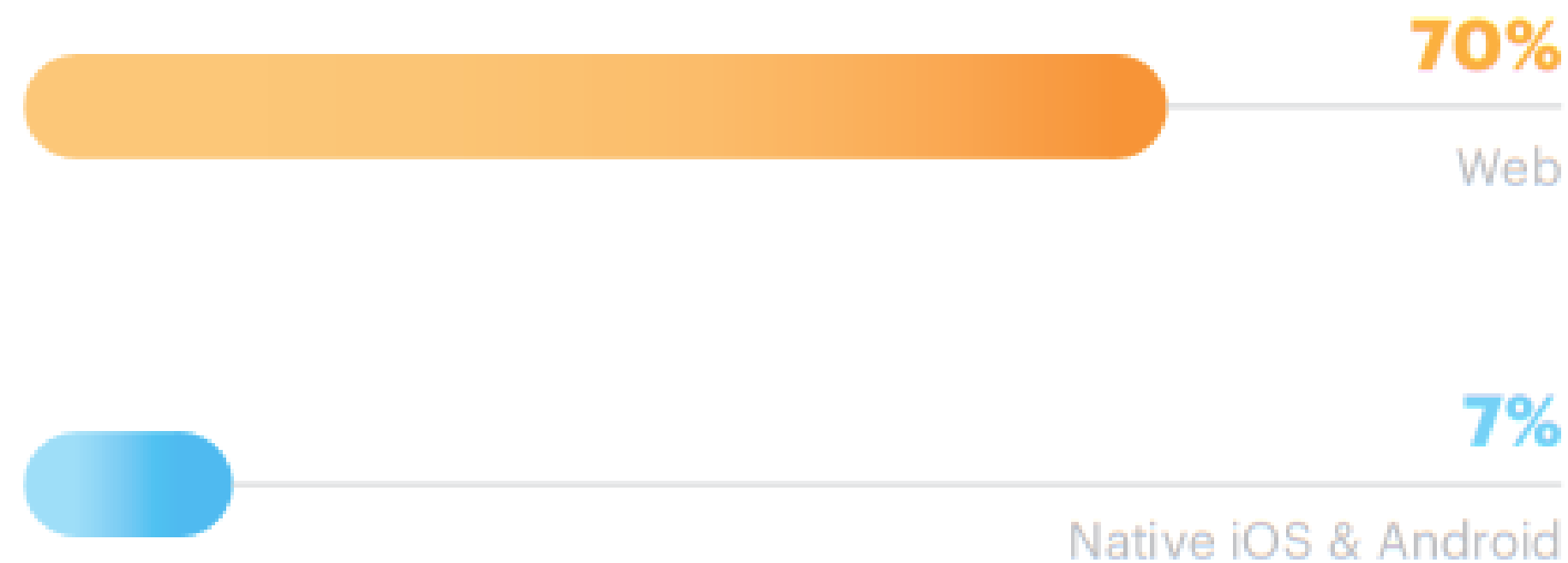


- » Rara vez es una aplicación móvil diseñada solo para una sola plataforma.
- » Los consumidores, socios y empleados tienen una variedad de plataformas y dispositivos.

android 

ios

- » La comunidad Web es aproximadamente 10 veces mayor que la comunidad nativa. ¿Quién no ha hecho páginas web con HTML, CSS y JS?



According to the 2019 Stack Overflow Survey, less than 7% of all developers cited Swift, Kotlin, or Objective-C as familiar languages. In contrast, web developers made up almost 70% of respondents.

- » Savello App tiene 3M+ de usuarios con una puntuación de 4.8 de 5 estrellas y esto se debe gran parte por la experiencia del usuario al usar la aplicación

4.8



With over 3 million users

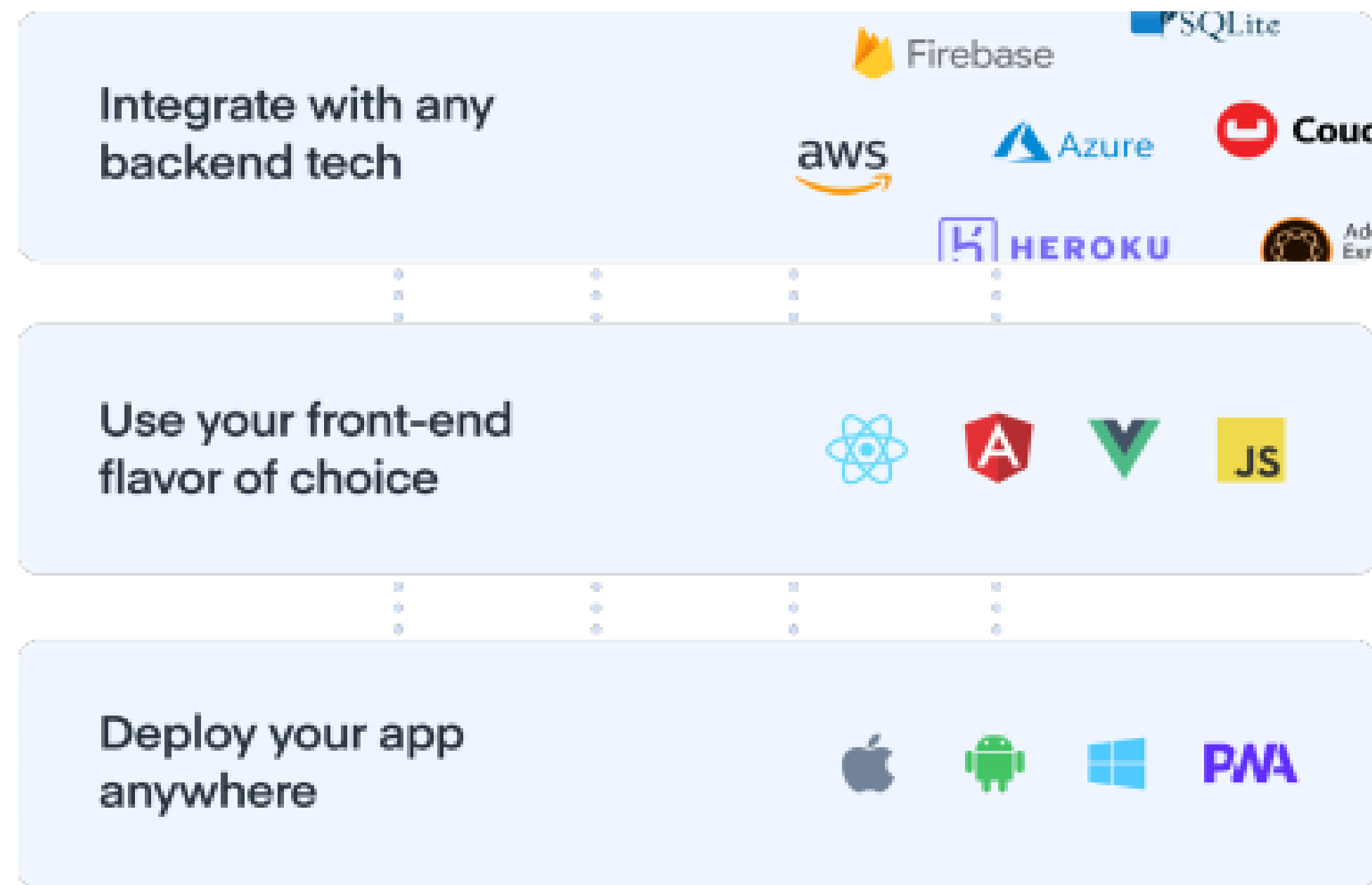


SANVELLO APP



PGY4121 | Integraciones

» Desarrollo totalmente compatible con integraciones.



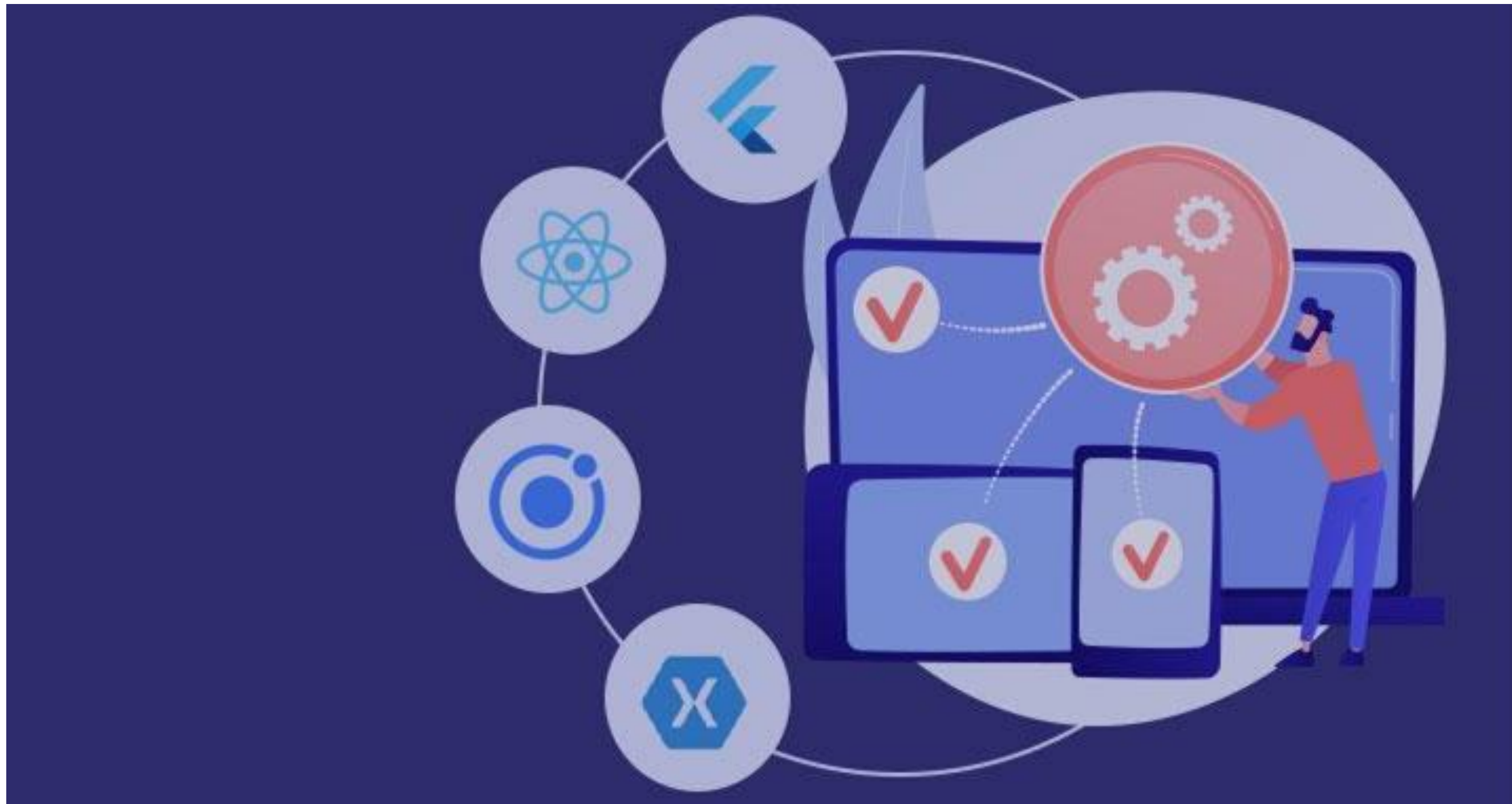
PGY4121 | ¿Por qué Nativo?

- » Rendimiento mejor que el Híbrido
- » Completa librería nativa
- » Sin dependencia de Terceros



PGY4121 | Frameworks

» Para el desarrollo Nativo tenemos diferentes Frameworks

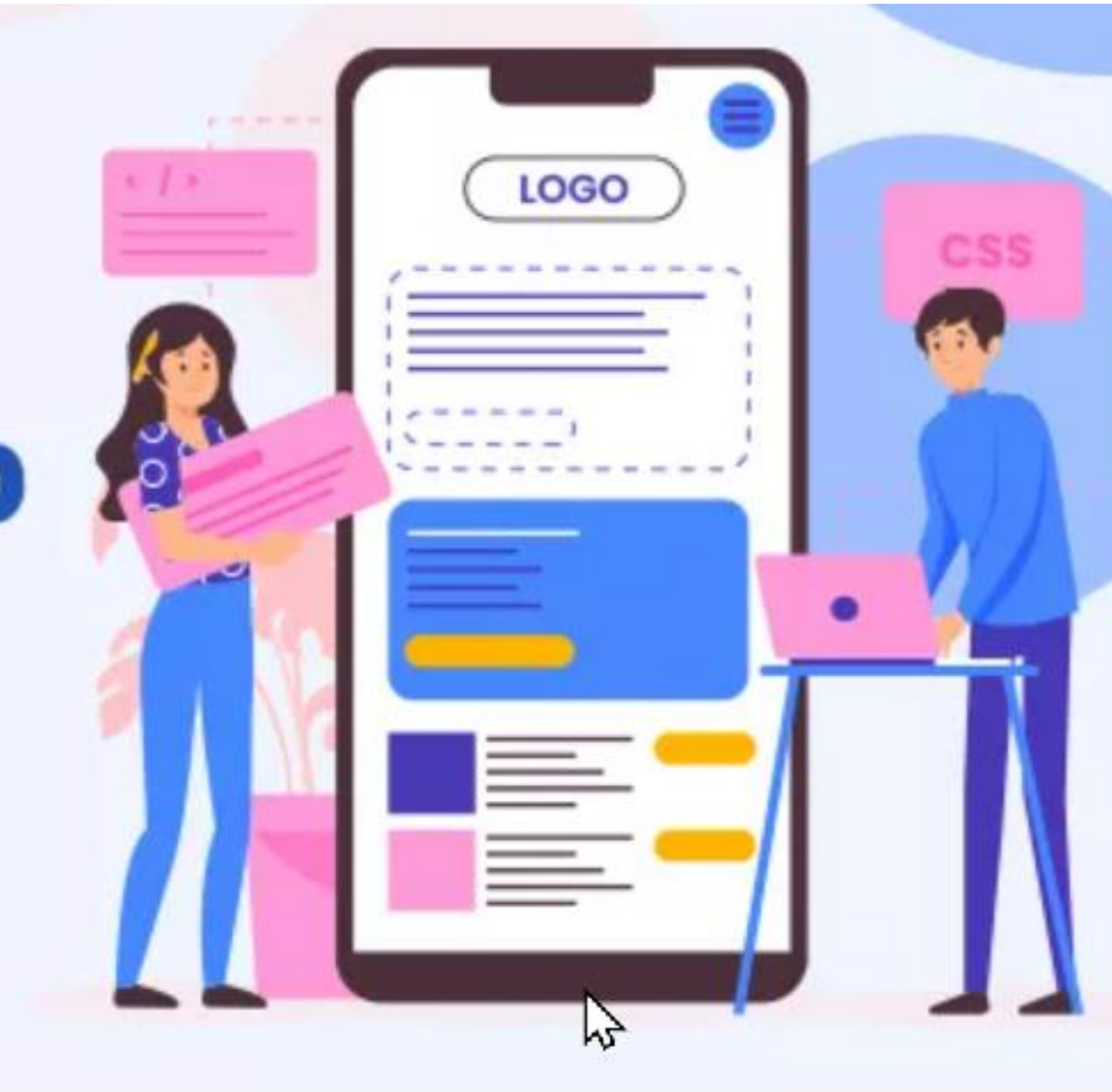


Ranking Frameworks más utilizados - Información extraída en Diciembre 2022

» Para el desarrollo Nativo tenemos diferentes Frameworks

5

Best Hybrid App Frameworks for 2022



Flutter

Flutter is an efficient Mobile UI framework developed by Google. With flutter one can craft a high-quality native interface which is compatible for Android and iOS.



React Native

Open-source, Reusable Components. Developed by Facebook developer. Used by Facebook, Instagram, Uber Food etc. easily integrate native feature with Hybrid.



Ionic

Open Source SDK, Offers all of the functionalities like native mobile development SDKs.



Xamarin

Based on .NET Framework for iOS, Android, and Windows. Acquired by Microsoft in 2016 and made SDK open-source, and it became part of Xamarin Visual Studio IDE.

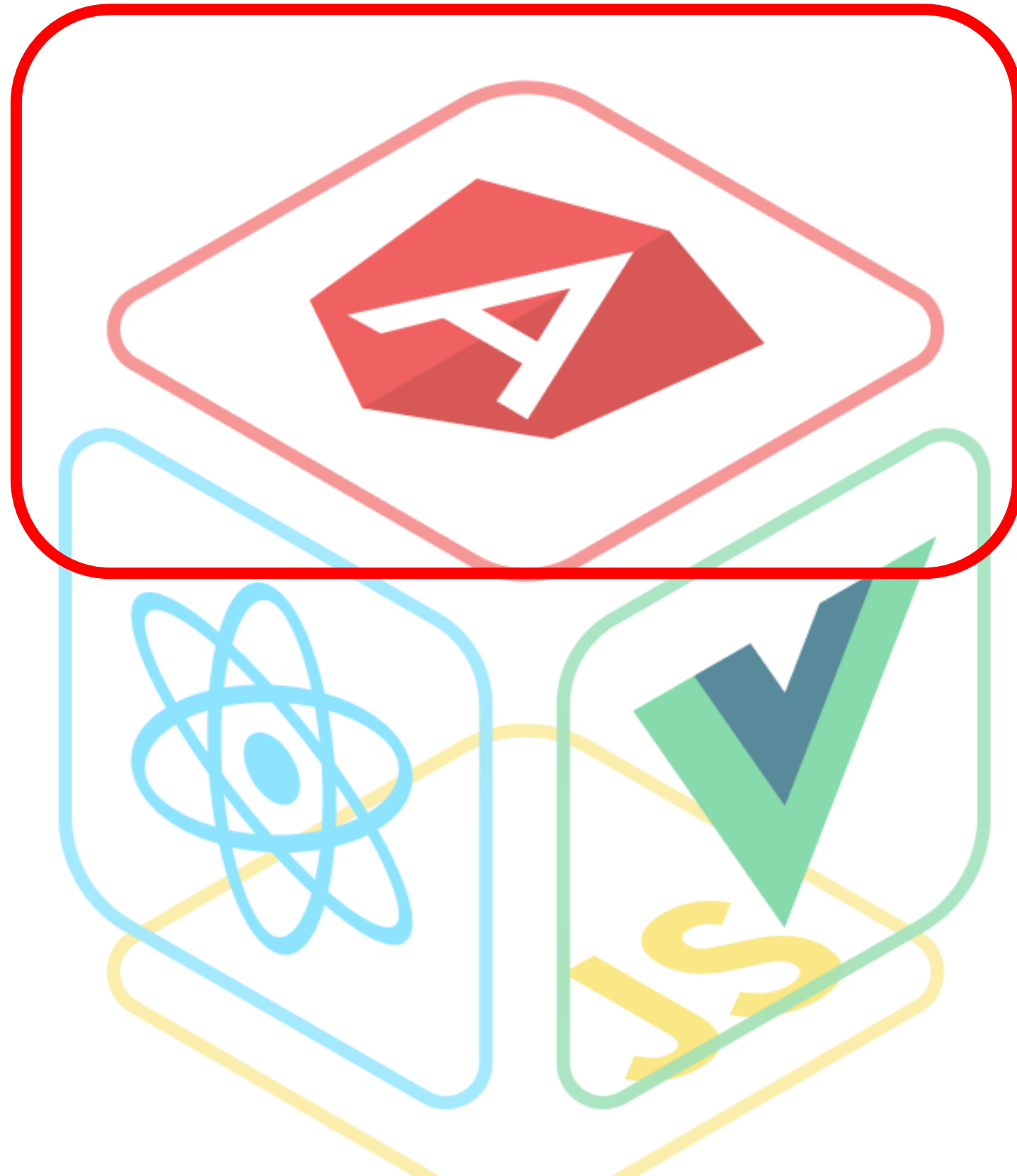


Framework 7

No external dependency & A Lots of UI elements and widgets.

PGY4121 | Ionic Angular

» Nosotros en esa asignatura usaremos:



Ionic Angular



PGY4121

Composición Page

» nombre_page-routing.module.ts

▼ home

TS home-routing.module.ts

TS home.module.ts

home.page.html

home.page.scss

TS home.page.spec.ts

TS home.page.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomePage } from '../home.page';

const routes: Routes = [
  {
    path: '',
    component: HomePage,
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})

export class HomePageRoutingModule {}
```


» nombre_page.module.ts

✓ home

TS home-routing.module.ts

TS home.module.ts

home.page.html

home.page.scss

TS home.page.spec.ts

TS home.page.ts

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { IonicModule } from '@ionic/angular';
import { FormsModule } from '@angular/forms';
import { HomePage } from '../home.page';
import { HomePageRoutingModule } from '../home-routing.module';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    HomePageRoutingModule
  ],
  declarations: [HomePage]
})
export class HomePageModule {}
```

PGY4121 | Composición Estructural

» nombre_page.page.html

✓ home

TS home-routing.module.ts

TS home.module.ts

home.page.html

home.page.scss

TS home.page.spec.ts

TS home.page.ts

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title> Blank </ion-title>
  </ion-toolbar>
</ion-header>
<ion-content [fullscreen]="true">
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">Blank</ion-title>
    </ion-toolbar>
  </ion-header>
  <div id="container">
    <strong>Ready to create an app?</strong>
    <p>Start with Ionic <a target="_blank" rel="noopener noreferrer"
href="https://ionicframework.com/docs/components">UI Components</a></p>
  </div>
</ion-content>
```


» nombre_page.page.scss

✓ home

TS home-routing.module.ts

TS home.module.ts

home.page.html

home.page.scss

TS home.page.spec.ts

TS home.page.ts

```
#container {
  text-align: center;
  position: absolute;
  left: 0;
  right: 0;
  top: 50%;
  transform: translateY(-50%);
}

#container strong {
  font-size: 20px;
  line-height: 26px;
}
```

» nombre_page.page.ts

▼ home

TS home-routing.module.ts

TS home.module.ts

home.page.html

home.page.scss

TS home.page.spec.ts

TS home.page.ts

```
import { Component } from '@angular/core';
```

```
@Component({  
  selector: 'app-home',  
  templateUrl: 'home.page.html',  
  styleUrls: ['home.page.scss'],  
})
```

```
export class HomePage {
```

```
  constructor() {}
```

```
}
```

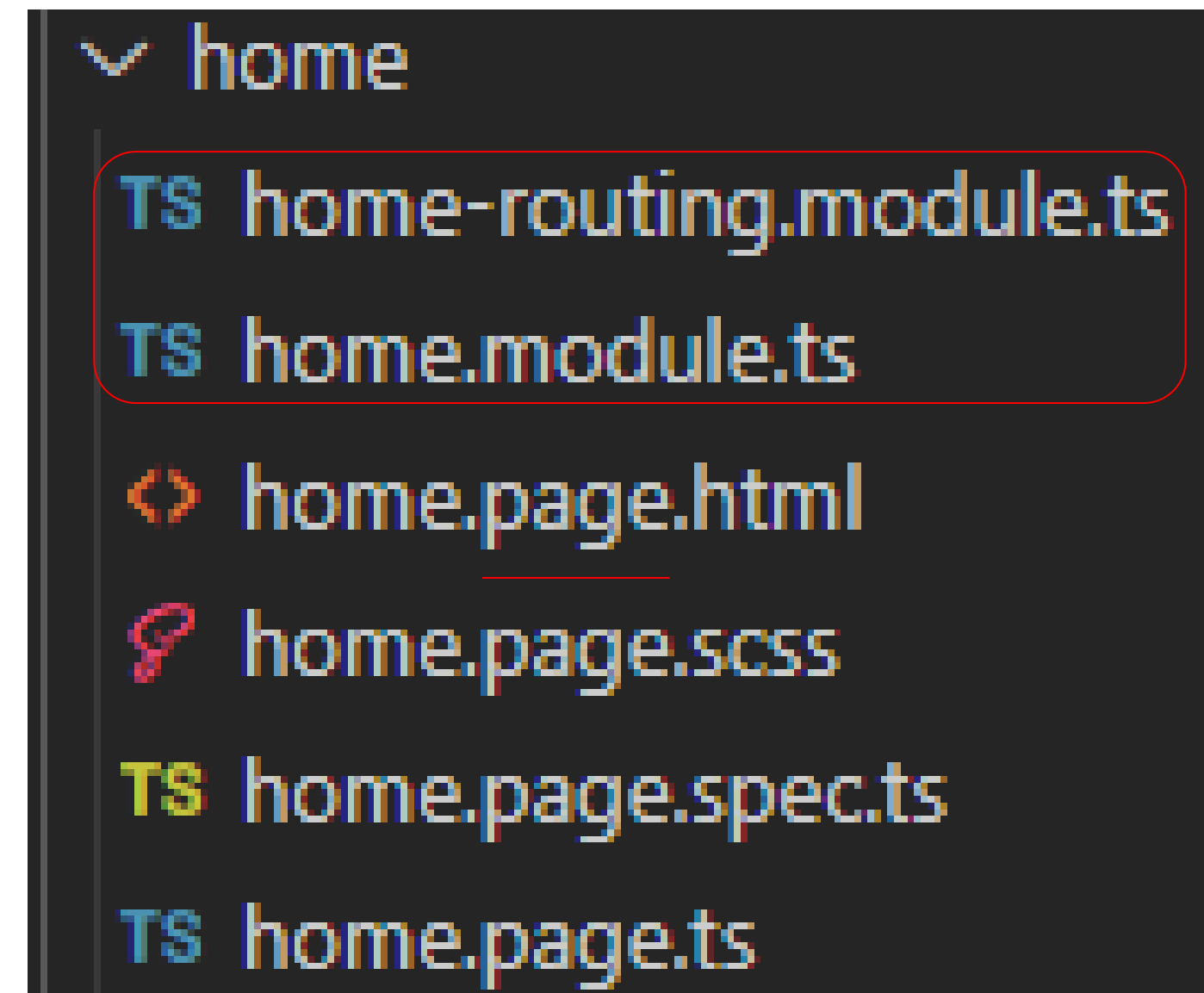
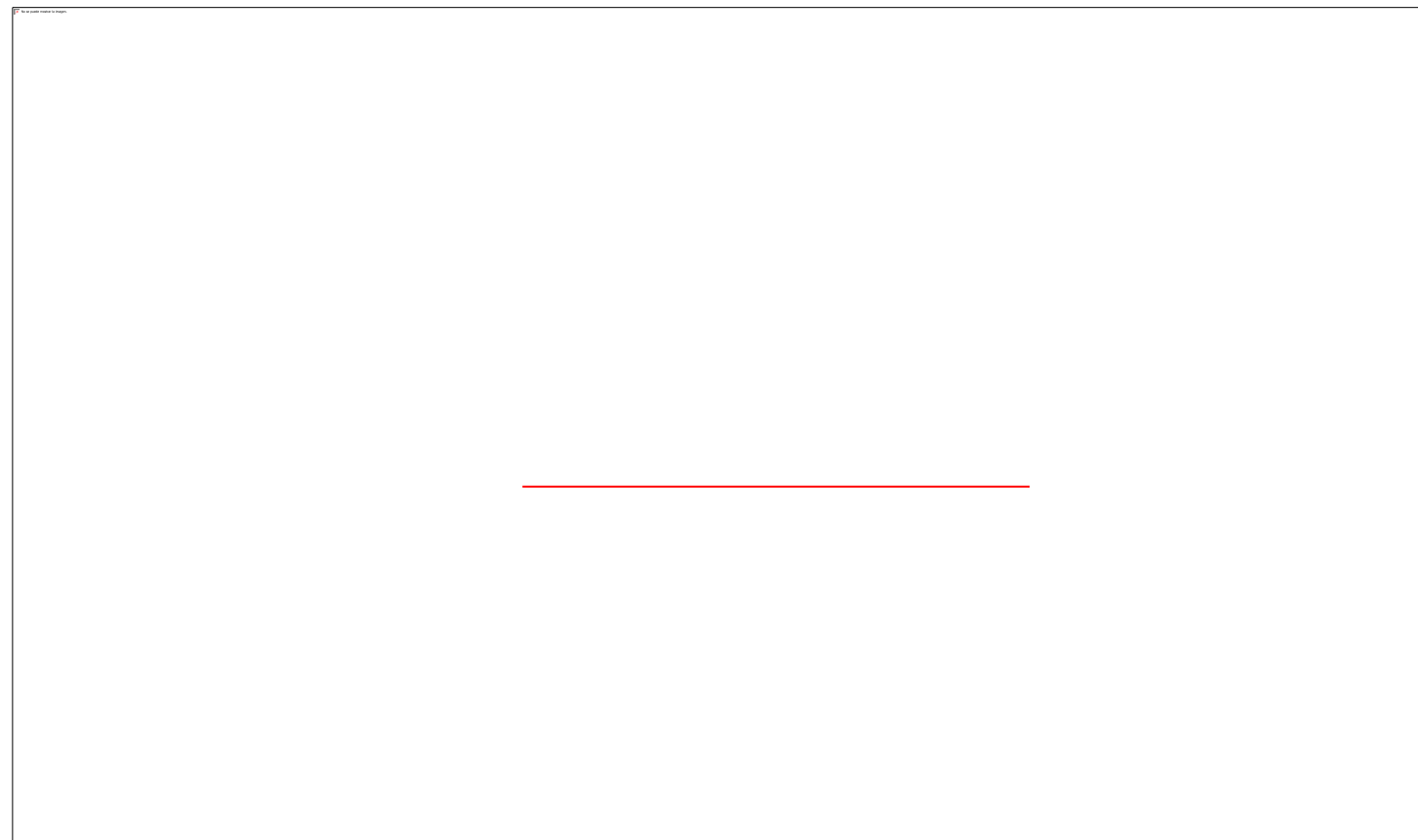



PGY4121

Composición Component

PGY4121 | Composición Estructural

- » El componente posee casi la misma estructura que un Page con la salvedad que no posee un `nombre_page.module.ts` ni un `nombre_page-routing.module.ts` y su sufijo es `component`





Documentación Oficial

» Para esta asignatura se les recomienda consultar la documentación oficial <https://ionicframework.com/docs>