

Sesión Sincrónica Nº 3

Programación de aplicaciones móviles (PGY4221)




Profesor: Miguel Puebla

Experiencia 1: Framework híbrido

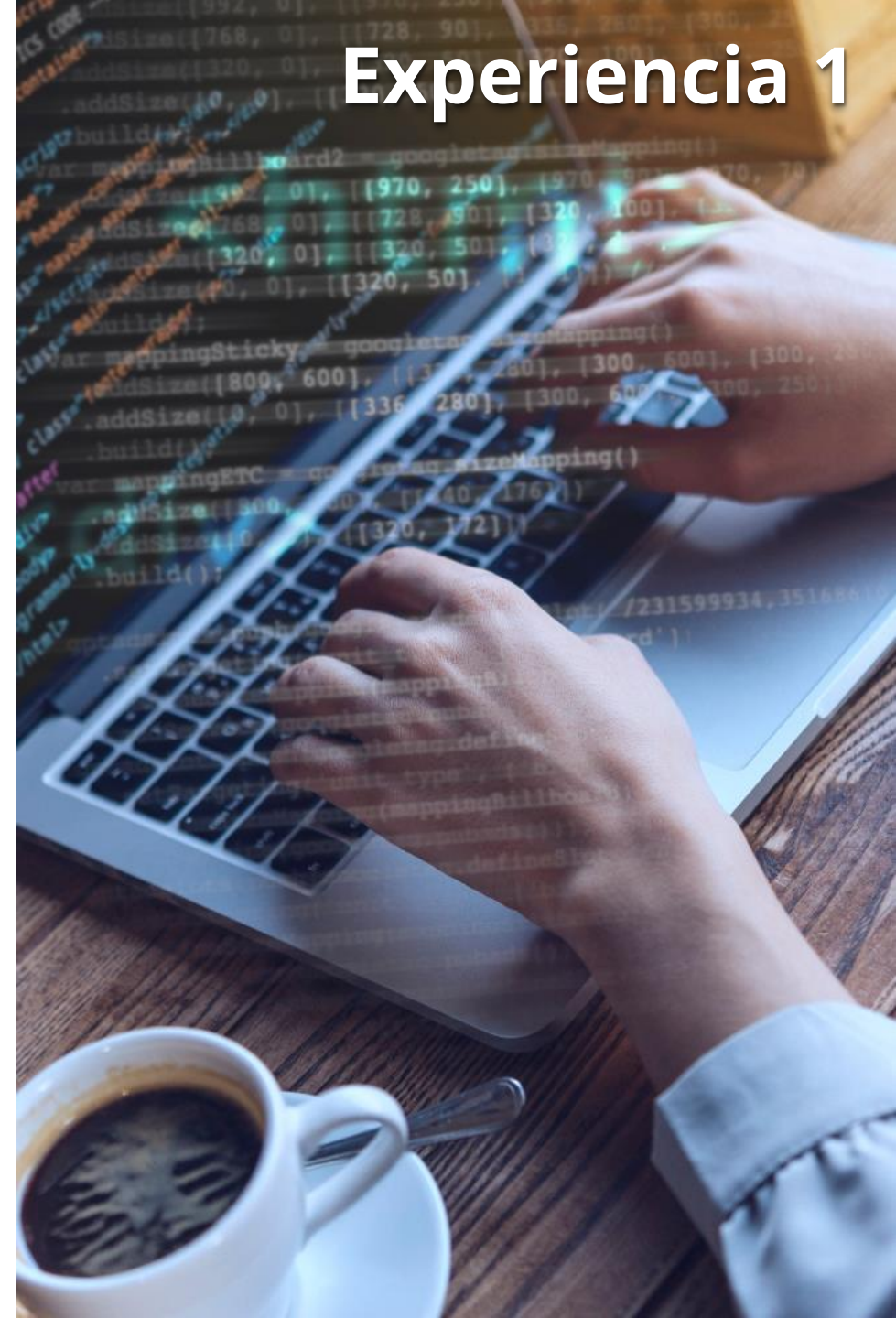
RA1. Crea una primera versión básica de una aplicación móvil utilizando un framework y con un diseño creativo y funcional que permita el desarrollo de interfaces gráficas coherentes y de fácil uso.

RA2. Desarrolla una nueva versión híbrida de la aplicación móvil, considerando el uso de prestaciones, el manejo de interfaces y API's que permitan la optimización de códigos y un correcto desarrollo de sus funciones.

Semana 3: Creando mi Primera App

-  **IL2.** Desarrolla interfaz de usuario del framework coherente con los componentes respectivos y su animación.
-  **IL3.** Construye la navegación de la aplicación, siguiendo los pasos establecidos para alcanzar su correcto funcionamiento.
-  **IL4.** Mejora la usabilidad y las prestaciones del equipo a través del enrutamiento en la navegación de la aplicación conforme a los requerimientos del cliente.

Experiencia 1



Semana 3

“Creando mi Primera App”



Conocimientos Generales

Desarrollo móvil híbrido	Framework Ionic	Angular
Ionic Generate		



Desarrollo móvil híbrido

Los componentes de UI en Ionic son elementos preconstruidos que permiten crear interfaces de usuario rápidas y consistentes para aplicaciones móviles, web y de escritorio. Están diseñados para funcionar en cualquier plataforma y se adaptan al estilo nativo de la plataforma en la que se ejecutan, como iOS, Android o la web.

Características del Desarrollo Móvil Híbrido

Característica	Descripción
Tecnologías	HTML, CSS, JavaScript
Frameworks populares	Ionic, React Native, Flutter
Uso de plugins	Permite acceso a funcionalidades nativas del dispositivo (cámara, GPS) a través de plugins como Apache Cordova
Mantenimiento	Fácil de mantener, ya que usa un solo código base
Experiencia de usuario	Similar a la de apps nativas, aunque puede variar según el dispositivo
Tiempo de desarrollo	Reducido en comparación con apps nativas, ya que permite reutilizar código
Desempeño	Bueno, aunque puede ser menor que una app nativa en funcionalidades que requieren alta eficiencia
Compatibilidad multiplataforma	Compatible con iOS, Android, y web (PWA)

Desarrollo móvil híbrido

Comparación entre Aplicaciones Híbridas y Nativas

Característica	Aplicación Nativa	Aplicación Híbrida
Código base	Uno para cada plataforma (Android, iOS)	Único código para todas las plataformas
Tecnologías de desarrollo	Lenguajes específicos (Kotlin, Swift)	HTML, CSS, JavaScript
Rendimiento	Superior en rendimiento y acceso a hardware	Bueno, pero inferior en casos de alto rendimiento
Experiencia de usuario	Altamente optimizada	Puede variar; depende de la calidad del framework
Tiempo de desarrollo	Más largo debido a la necesidad de crear apps separadas	Más corto; una sola base para todas las plataformas
Acceso a funcionalidades	Completo y optimizado	Limitado, aunque puede acceder a funciones nativas con plugins
Costos	Generalmente más altos	Más bajo debido a la reutilización de código

Desarrollo móvil híbrido

Enfoque en Ionic

Ionic es uno de los frameworks más utilizados para el desarrollo móvil híbrido. Basado en tecnologías como **Angular**, React y Vue, facilita la creación de interfaces atractivas y altamente funcionales. Ionic permite acceder a características nativas del dispositivo mediante Capacitor (su propio motor de plugins) o Cordova, proporcionando una experiencia cercana a la de las aplicaciones nativas.





Framework Ionic

Características del Framework Ionic

Característica	Descripción
Lenguajes y Tecnologías	Basado en HTML, CSS y JavaScript; soporta frameworks como Angular, React y Vue para estructurar aplicaciones
Interfaz de Usuario (UI)	Incluye una extensa librería de componentes de UI pre-diseñados (botones, listas, formularios, íconos) que emulan el estilo nativo
Motor de Plugins	Usa Capacitor o Cordova para integrar funciones nativas (cámara, GPS, almacenamiento) de forma sencilla
Compatibilidad multiplataforma	Permite la creación de aplicaciones para Android, iOS, y Progressive Web Apps (PWA) usando un único código base
Rendimiento	Optimizado para maximizar el rendimiento, aunque puede ser menos eficiente que apps nativas en procesos intensivos
Fácil instalación y despliegue	Su CLI (Command Line Interface) facilita la creación de proyectos, generación de páginas, y ejecución en emuladores
Personalización	Altamente personalizable gracias a CSS, facilitando el ajuste de estilos visuales según las necesidades de la aplicación
Comunidad y soporte	Posee una gran comunidad y documentación detallada, así como soporte en foros, tutoriales y herramientas adicionales de Ionic Pro
Actualizaciones frecuentes	El equipo de Ionic proporciona actualizaciones constantes y mejoras que incluyen compatibilidad con las últimas versiones de Angular, React, y Vue
Integración con otros servicios	Compatible con Firebase, Stripe, Google Maps, y otras APIs y SDKs de terceros, facilitando la integración de servicios adicionales en la app

Angular

Angular es un framework de desarrollo frontend creado por Google, diseñado para construir aplicaciones web dinámicas y escalables. Utiliza TypeScript, permitiendo una estructura modular basada en componentes, que facilita la creación, prueba y mantenimiento de aplicaciones complejas. Angular ofrece herramientas avanzadas como inyección de dependencias, enlace bidireccional de datos y un enrutador poderoso, además de una CLI que agiliza el desarrollo y optimización del código.

HTML

```
<h2>Lista de Tareas</h2>
<input [(ngModel)]="nuevaTarea" placeholder="Nueva tarea">
<button (click)="agregarTarea()">Agregar</button>
<ul>
  <li *ngFor="let tarea of tareas">{{ tarea }}</li>
</ul>
```

TypeScript

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

@Component({
  selector: 'app-todo',
  templateUrl: './todo.component.html',
  styleUrls: ['./todo.component.css']
})
export class TodoComponent {
  tareas: string[] = [];
  nuevaTarea: string = '';

  agregarTarea() {
    if (this.nuevaTarea) {
      this.tareas.push(this.nuevaTarea);
      this.nuevaTarea = '';
    }
  }
}
```


Angular

Características del Framework Angular

Característica	Descripción
Lenguaje principal	TypeScript, un superconjunto de JavaScript que agrega tipos estáticos y herramientas de desarrollo avanzadas
Arquitectura basada en componentes	Facilita la construcción de aplicaciones complejas y escalables mediante componentes modulares y reutilizables
Data Binding (Enlace de datos)	Incluye enlace bidireccional de datos, permitiendo la sincronización automática entre la vista y el modelo
Inyección de dependencias (DI)	Angular cuenta con un sistema de DI que permite una fácil administración y prueba de las dependencias
Modularidad	La estructura modular permite organizar y cargar módulos específicos para mejorar la eficiencia y escalabilidad
Ruteo y Navegación	Incluye un enrutador poderoso para definir rutas, facilitando la navegación en Single Page Applications (SPAs)
Soporte para formularios	Maneja formularios complejos con validaciones de manera sencilla, ofreciendo un sistema robusto para formularios reactivos
CLI (Command Line Interface)	La CLI de Angular permite generar componentes, servicios, módulos, y facilita el despliegue y la optimización de la aplicación
Actualizaciones periódicas	Recibe actualizaciones periódicas del equipo de Google, manteniendo la compatibilidad y mejoras de rendimiento
Integración con Ionic	Angular es compatible con Ionic, lo que permite construir aplicaciones móviles híbridas usando la misma base de código de Angular

Ionic Generate

Ionic Generate es un comando de la CLI de Ionic que permite crear automáticamente distintos elementos para una aplicación (páginas, componentes, servicios, entre otros) mediante plantillas predefinidas. Este comando simplifica la creación de archivos y estructuras, promoviendo una organización clara y una mayor rapidez en el desarrollo de aplicaciones.

Opciones de ionic generate

Comando	Descripción	Ejemplo
<code>ionic generate page</code>	Genera una nueva página con archivos asociados para HTML, CSS y TypeScript	<code>ionic generate page Home</code>
<code>ionic generate component</code>	Crea un componente reutilizable que se puede incluir en varias páginas	<code>ionic generate component Header</code>
<code>ionic generate service</code>	Genera un servicio para manejar lógica de negocio y facilitar la comunicación entre componentes	<code>ionic generate service Data</code>
<code>ionic generate module</code>	Crea un módulo para organizar el código y facilitar el lazy loading	<code>ionic generate module SharedModule</code>
<code>ionic generate directive</code>	Crea una directiva personalizada que agrega comportamiento especial a elementos HTML	<code>ionic generate directive Highlight</code>
<code>ionic generate guard</code>	Crea un guard para proteger rutas en función de condiciones definidas	<code>ionic generate guard AuthGuard</code>
<code>ionic generate pipe</code>	Genera un pipe para transformar datos en las vistas	<code>ionic generate pipe FormatDate</code>
<code>ionic generate enum</code>	Crea una enumeración, útil para definir conjuntos de valores constantes	<code>ionic generate enum UserRole</code>

Ejercicios



Lecturas



Phan, H., Singh, I. (2018). Ionic Cookbook: Recipes to create cutting-edge, real-time hybrid mobile apps with Ionic, 3rd Edition. Packt Publishing.

¡Muchas Gracias!



DuocUC[®] 
ONLINE