

# INSTITUTO SUPERIOR POLITÉCNICO CÓRDOBA

## TECNICATURA SUPERIOR EN DESARROLLO WEB Y APLICACIONES DIGITALES

### EXAMEN FINAL PARA ESTUDIANTES LIBRES COHORTE 2023

Este documento está dirigido a los/as estudiantes que deben rendir en mesa examinadora final del Módulo Aplicaciones para móviles en **condición de LIBRE**.

- **Formato de la Práctica Profesionalizante/Formativa/ABP**

Elaboración de Proyectos de software

- **Objetivos**

Elaborar un proyecto con el fin de lograr que los estudiantes demuestren consolidar, integrar y/o ampliar las capacidades y saberes que corresponden al Programador de Aplicaciones móviles.

Desarrollar una **Aplicación Móvil** funcional implementada para Sistema Operativo Android aplicando buenas prácticas de Ciberseguridad y Testing e integrándolo dentro de una gestión de Proyecto de Software

- **Lineamientos Generales**

- Planificación
- Análisis
- Diseño
- Programación
- Pruebas
- Implementación

- Mantenimiento
- Documentación

La temática de la aplicación queda a criterio del estudiante.

La aplicación deberá implementar el contenido mínimo dictado en clases sincrónicas, asincrónicas y material de la plataforma pudiendo anexar lo que crea necesario justificándose; y todos los requerimientos necesarios para la organización de trabajo en equipo, simulando el trabajo en una empresa real, aunque el trabajo sea individual. A saber: todos los puntos considerados en el apartado CONSIGNAS.

Se recomienda utilizar IDE (Entorno de Desarrollo Integrado) utilizados al cursar cada espacio, a saber:

1. Android Studio: desarrollo de la Aplicación Móvil con JAVA.
2. Al momento de presentarse se debe poder visualizar todos los aspectos de la aplicación: código, diseño

## • Criterios de Evaluación

El estudiante deberá presentarse el día del examen con la aplicación móvil y el proyecto en su totalidad en el repositorio de trabajo, desarrollados de acuerdo a los ítems designados por los Docentes.

Debiendo realizar la defensa del trabajo en el espacio disponible para videoconferencia que presenta la plataforma (BBB/Meet) en la correspondiente mesa examinadora.

La defensa del trabajo consiste en que el/la estudiante muestre el desarrollo de la aplicación en el entorno de trabajo para que se pueda apreciar: la estructura del proyecto, el código y los aspectos de diseño y recursos utilizados. Posteriormente se procederá a ejecutar la aplicación y se realizan algunas preguntas técnicas sobre lo solicitado en el proyecto.

Entrega a término (todos los entregables solicitados requeridos enumerados en el apartado evidencias requeridas)

- Aplicación funcional, no serán válidos proyectos que presenten problemas de compilación, o ejecución al momento de la presentación.

- Código ejecutable sin errores, no se recibirán archivos con errores ni de sintaxis ni de lógica. (si la aplicación en el IDE no compila no puede ser evaluada)
- Detalle de la evidencia del proceso completo.
- Detalle de errores en los que se incurrió y pudo resolverlo.
- Utilización de buenas prácticas al codificar
- Aplicación de conceptos de ciberseguridad alineados a las buenas prácticas de protección de posibles amenazas.
- Implementación de un Plan de Gestión de Riesgos basado en Políticas de Seguridad detallado en la Normativa Internacional ISO 27001 y 27002
- Desarrollo de casos de prueba de calidad que permitan a la aplicación estar exenta de incidentes de alto riesgo comparando resultados deseados con resultados esperados.
- Lenguaje técnico.

Todos deben tener un proceso de registro/login/CRUD, con persistencia de datos.

## • Consignas

Funcionalidades completas para crear una APP:

- El inicio de la aplicación deberá estar bien definida.
- Registro de usuarios, y login de usuarios verificando carga en BD.
- Gestión CRUD básico funcional. Ej: producto/servicio/cliente/etc. (deberá explicar el código básico necesario para esta implementación)
- La comunicación de la aplicación con la Base de Datos se debe realizar a través de una API en Django. Como ésto se realiza en la estación de trabajo del estudiante, la API puede implementarse de manera local sin necesidad de desplegarlo en la nube.
- Realizar traspaso de datos entre Activitys
- Incluir elemento multimedia. Ej: Video
- Incluir imágenes o recursos en el diseño de las activitys

La aplicación deberá contar con un esquema de navegación entre las distintas activitys que permita al usuario volver a un punto principal y no presente en ningún caso lugares sin retorno.

***Se deberán entregar de manera requerida las siguientes evidencias:***

- 1) Exportar el proyecto desde el IDE utilizado a un **archivo .zip** sin errores ni de sintaxis, ni de lógica en el código.
- 2) Un archivo de volcado de la base de datos que contenga estructura y contenido, con el que se puedan realizar copias de seguridad para pruebas y de ésta forma garantizar la validez, la coherencia y la capacidad de manipular la información.
- 3) Especificar el IDE y la versión con la que se realizó el proyecto.
- 4) **Tener todo documentado** en la Wiki y/o enlaces a Drive el proceso completo (relevamiento de necesidades, requerimientos del cliente hasta ejecución) acompañado de capturas de pantalla, trozos de código y toda evidencia que crea fundamental para que se comprenda de manera clara el objetivo y los pasos secuenciales de inicio a fin que se realizaron para el resultado final.
- 5) Diseño y ejecución de al menos **5 casos de prueba** donde se pueda observar el **resultado esperado, datos y pasos** de las funcionalidades de la aplicación (que se deriva de los requerimientos / criterios de aceptación) y el **resultado obtenido** del mismo. Si hubiera incidente indicar el grado de riesgo y especificar cómo se subsanó (evidencia) en su repositorio GitHub. El detalle del mismo se incluirá en el detalle del apartado 4.
- 6) Documentar 5 *pruebas de accesibilidad* en base a la Ley N° 26.653 de "Accesibilidad Web", facilitando especialmente el acceso a todas las personas con discapacidad con el objeto de garantizarles la igualdad real de oportunidades y trato, evitando así todo tipo de discriminación.
- 7) Formular Políticas de Seguridad para definir el Sistema de Gestión de Riesgos de la Aplicación Móvil, asignando la frecuencia en la cual debe ejecutarse/revisarse y asignar uno o más responsables para cada una.
- 8) Interrelación Ciberseguridad y Proyecto: Relacionar Requerimientos de Usuario con Políticas de Seguridad, que Política el Usuario no va a solicitar pero se debe implementar como buena Práctica Profesional de Ciberseguridad.
- 9) Interrelación Ciberseguridad y Programación: Desarrollar la autenticación de la aplicación con Auth0 o JWT.
- 10) Implementar autorización según perfiles y permisos de usuarios.
- 11) Interrelación Ciberseguridad y Testing: Planificar al menos 3 casos de prueba basados en políticas de seguridad de acuerdo a la plantilla de test Case propuesta en Testing. Asignar frecuencia y responsables de prueba.
- 12) OPCIONAL: Implementar buenas prácticas y legislación presentadas en la asignatura: botón de arrepentimiento, ley de protección de datos personales, aceptación de términos y condiciones, entre otros.

13) Un **video corto enviado con antelación (opcional)** donde el estudiante evidencie y explique con lenguaje técnico la aplicación funcionando y el código (clase, Activity, eventos) de manera alternada de forma tal que se pueda visibilizar el proceso que él mismo hizo codificando para obtener el resultado, así como también, se aplicaron políticas de seguridad que protegen a la aplicación y como se testeó con el fin de que el código no posea incidentes de alto riesgo como se describió en el punto 6. ***(El objetivo del Video es poder tener más en claro el proceso del desarrollo y lo aplicado en el proyecto, de lo contrario tendría que librarlo a incluirlo en el día del examen)***

14) El estudiante deberá poder explicar la modularidad de los 4 espacios, relacionar como se planifica el proceso de desarrollo de la aplicación, como es que de esa planificación se derivan políticas de seguridad, como se codifican y como se testean de manera transversal. No probar lo que no se planificó ni se implementó, todo tiene que tener su correlación.

15) El estudiante deberá proporcionar acceso a todos los entregables solicitados en los puntos anteriores (ejemplo: subir a un directorio y compartirlo)

### **Proyecto Integrador – Se valora el MAIN**

- Logra conectarse al repositorio GitHub (colaboradores/contribuidores).
- Logra agregar **Issues como Historia de Usuarios /tareas**, (mínimo 10 issues) asignarlos a colaboradores y en **Milestones por Sprint**, (sprints con fechas de inicio y final, se sugiere sprint 0, 1 y 2) respetando la nomenclatura de las issues sugeridas. ***No olvidar los criterios de aceptación (importantes para el momento de realizar el testing y ciberseguridad).***
- Logra realizar **Commit** en todo el recorrido de Sprints, en las ramas feature, dev. (No se considerarán los que digan “add file upload”)
- Logra crear **Wiki** con registro de ceremonias (daily, review, retrospective) y novedades del equipo.
- Utiliza **kanban** (Product Backlog, ToDo, In process, Finished).
- Logra Crear/Actualizar **documentación IEEE830**
- Logra mergearear /pull al main actualizando todo el proyecto.
- Logra diseñar el modelo de datos conceptual (DER): Entidad, Atributos y cardinalidad.(imagen/pdf)
- Logra diseñar el modelo de datos lógico (Modelo Relacional) a partir del DER: (debe estar mínimo en 3ra forma normal) (imagen/pdf)

- Logra diseñar el diagrama de clases UML: Relaciones.(imagen/pdf)

Sintetizar el recorrido por la gestión del proyecto en GitHub.

## CONSIDERACIONES FINALES

- El proyecto puede ser presentado con anterioridad para revisión, enviando el link (enlace) del repositorio GitHub: **LOS ENVÍOS SE RECIBIRÁN HASTA 96 hs antes de la MESA EXAMINADORA.**
- La modalidad de trabajo en esta instancia es individual.
- Cada desarrollador debe conocer por completo el código de cómo fue creada la aplicación.
- En el momento del examen, **cada uno** debe tener todos los recursos necesarios para poder realizar la demostración integral del proyecto: Cámara, micrófono, Android Studio funcionando desde emulador (en PC o Celular) para compartir pantalla. TIEMPO DE TOLERANCIA 15 min. Se recomienda tener instalado Google Chrome ó Mozilla Firefox para poder compartir pantalla.

## CONSIDERACIONES PARA DIA DE LA MESA EVALUADORA

**A** la hora de correr la aplicación, puede por ejemplo hacerlo desde un teléfono físico (compartiendo pantalla en el aula), teniendo esto como última opción **solo para alumnos que demuestren que no cuentan con un equipo con los requerimientos mínimos suficientes para emular un dispositivo.**

**Preguntas orientadoras en el momento del examen, que el estudiante mencione:**

¿Dónde se hace el CRUD completo?

¿Usan APIS externas? Especificar

Esperamos que el estudiante sepa ¿cómo se usan las APIS? etc, etc, etc.

El estudiante debe brindar evidencias de que conoce el código, que sabe programar, que tiene conocimiento sobre el backend y front end.

¿Qué tipo de base de datos utiliza? ¿Cómo es la conexión con la misma?

Que justifique las decisiones que se tomaron para emplear la misma.

¿Cómo se comunican las diferentes actividades?