|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **GUÍA DE LABORATORIO DE**  Haga clic o pulse aquí para escribir texto. | | | | | | |
| **Unidad Didáctica: 2** | | | | | | |
| **Eje Temático:** Programación Avanzado | | | | | | |
| **No. Guía** | | 11 | | **Resultados de Aprendizaje de la Unidad Didáctica:**  Desarrolla APPs con acceso a bases de datos locales y en la nube | | |
| **1**  **sesiones** | | **13-14 Semana** | |
| **Horas de Trabajo** | | | |
| **Trabajo con Docente** | | **Trabajo Autónomo** | |
| **4** | | **8** | |
| **Tipo de trabajo** | | | |
| **Grupal** |  | **Ind** |  | **Laboratorio Requerido** | **Sala de Cómputo** | |
| **Introducción** | | | | | | |
| **Sesión 1. ¿Qué haremos en la sesión? ¿Cómo lo haremos?**   1. Crear un proyecto Xamarin Forms agregando el Nuget de SqLite.  **Agregando soporte de SQLite** Para agregar soporte de *SQLite* a una biblioteca de clases .NET Standard podemos utilizar la versión más reciente del paquete NuGet ***sqlite-net-pcl.*** Esto nos permitirá incorporar operaciones de bases de datos en el código compartido haciendo referencia a las clases *SQLite* incluidas en el paquete NuGet. Las operaciones de base de datos pueden ser definidas en el proyecto de biblioteca .NET Standard de la solución Xamarin.Forms o en un proyecto biblioteca de clases .NET Standard independiente.    Hay varios paquetes NuGet con nombres similares, el paquete correcto tiene los siguientes atributos:   * Creado por: Frank A. Krueger * Id: sqlite-net-pcl * Vínculo de NuGet: sqlite-net-pcl (<https://www.nuget.org/packages/sqlite-net-pcl/>)   A pesar de que el nombre del paquete incluye el acrónimo PCL y que pareciera con ello estar diseñado para trabajar con bibliotecas PCL, podemos utilizar el paquete NuGet ***sqlite-net-pcl*** incluso en los proyectos .NET Standard.   Utilizando SQLite *SQLite* nos permite almacenar objetos en una base de datos. Por ejemplo, podríamos almacenar objetos de tipo ***Product***.      public class Product      {          [SQLite.PrimaryKey]          public int ID { get; set; }          public string ProductName { get; set; }          public decimal UnitPrice { get; set; }          public int UnitsInStock { get; set; }          public int CategoryID { get; set; }      }  Para realizar operaciones con la base de datos podríamos crear un repositorio con un constructor que se encargue de crear o abrir la base de datos en el directorio local de la plataforma.      public class ProductRepository      {          SQLite.SQLiteAsyncConnection Database;            public ProductRepository()          {  string DbFilePath = Path.Combine(                  Environment.GetFolderPath(                      Environment.SpecialFolder.LocalApplicationData),                  "NorthWind.db");              Database = new SQLiteAsyncConnection(DbFilePath);              Database.CreateTableAsync<Product>().Wait();          }      }  A continuación, podríamos agregar el código para realizar las operaciones CRUD.      public class ProductRepository      {          SQLite.SQLiteAsyncConnection Database;            public ProductRepository()          {              string DbFilePath = Path.Combine(                  Environment.GetFolderPath(                      Environment.SpecialFolder.LocalApplicationData),                  "NorthWind.db");              Database = new SQLiteAsyncConnection(DbFilePath);              Database.CreateTableAsync<Product>().Wait();          }    public Task<int> CreateProductAsync(Product product)          {              return Database.InsertAsync(product);          }            public Task<Product> GetProductByIDAsync(int id)          {              return Database.Table<Product>().Where(p => p.ID == id).FirstOrDefaultAsync();          }            public Task<int> UpdateProductAsync(Product product)          {              return Database.UpdateAsync(product);          }            public Task<int> DeleteProductAsync(Product product)          {              return Database.DeleteAsync(product);          }          public Task<List<Product>> GetProductsAsync()          {              return Database.Table<Product>().ToListAsync();          }            public Task<List<Product>> GetproductsByCategoryIDAsync(int ID)          {              return Database.QueryAsync<Product>(                  $"SELECT \* FROM [Product] WHERE [CategoryID] = {ID}");          }      }  El Repositorio estaría listo para poder ser utilizado en el proyecto Xamarin.Forms.  private void Create\_Clicked(object sender, EventArgs e)          {              var P = new Product              {                  ID = 1,                  ProductName = "Azucar",                  UnitPrice = 10,                  UnitsInStock = 7,                  CategoryID = 1              };              var Repository = new ProductRepository();              Repository.CreateProductAsync(P);          }              private async void Retrieve\_Clicked(object sender, EventArgs e)          {              var Repository = new ProductRepository();              var P = await Repository.GetProductByIDAsync(2);          }            private async void Update\_Clicked(object sender, EventArgs e)          {              var Repository = new ProductRepository();              var P = await Repository.GetProductByIDAsync(3);              P.ProductName = "\*" + P.ProductName;              await Repository.UpdateProductAsync(P);          }            private async void Delete\_Clicked(object sender, EventArgs e)          {              var Repository = new ProductRepository();              var P = await Repository.GetProductByIDAsync(1);              await Repository.DeleteProductAsync(P);          }            private async void GetAll\_Clicked(object sender, EventArgs e)          {              var Repository = new ProductRepository();              var Ps = await Repository.GetProductsAsync();          }            private async void GetByCategory\_Clicked(object sender, EventArgs e)          {              var Repository = new ProductRepository();              var Ps = await Repository.GetproductsByCategoryIDAsync(1);          }  **Sesión 2. ¿Qué haremos en la sesión? ¿Cómo lo haremos?**   1. Agregar la interfaz gráfica correspondiente para utilizar el repositorio del primer punto. 2. Enviar un documento .Pdf con la ejecución de la aplicación con el CRUD propuesto en el punto anterior 3. Pantallazo de la clase en la carpeta Models. 4. Pantallazo de la vista que invoca a cada método. | | | | | | |
| **Preguntas Orientadoras** | | | | | | |
| 1. Usted necesita desarrollar una aplicación para dispositivos móviles que utilizará gestión con bases de datos. La aplicación no se conectará a Internet y usted debe dar solución a este requerimiento.   ¿Qué procedimiento utilizará para activar el gestor de base de datos que reside en su celular?  **Para activar el gestor de base de datos en mi celular, primero debo asegurarme de que el gestor de base de datos esté instalado en mi dispositivo móvil. Si no está instalado, debo descargar e instalar una aplicación de gestión de bases de datos de mi elección desde la tienda de aplicaciones correspondiente.**  **Una vez que tengo el gestor de base de datos instalado, puedo crear una base de datos local en mi dispositivo móvil. Puedo hacer esto mediante la interfaz del gestor de base de datos, generalmente a través de la ejecución de comandos SQL. Luego puedo definir tablas, campos y relaciones en la base de datos.**  **Finalmente, para integrar la base de datos en mi aplicación, puedo utilizar una API de acceso a bases de datos local para interactuar con la base de datos y ejecutar consultas. De esta manera, puedo asegurarme de que mi aplicación utilice la base de datos local en lugar de conectarse a Internet para obtener o almacenar información.**   1. Revisando las tecnologías para gestión de información con apoyo en un SDBD nos damos cuenta que podemos utilizar herramientas en la web para nuestras aplicaciones móviles que permiten realizar estos procesos. ¿Cuáles tecnologías incorporaría a su desarrollo?   **Algunas tecnologías que podría incorporar en mi desarrollo para gestionar información con apoyo en un SDBD podrían ser:**  **SQLite: Es una biblioteca en C que implementa un SGBD SQL de pequeño tamaño y rápida ejecución. SQLite es una excelente opción para aplicaciones móviles debido a su simplicidad y eficiencia.**  **Firebase Realtime Database: Firebase es una plataforma móvil y web que proporciona herramientas y servicios para el desarrollo de aplicaciones móviles. Firebase Realtime Database es una base de datos en tiempo real alojada en la nube que se sincroniza en tiempo real con los clientes de aplicaciones. Firebase también ofrece otras herramientas, como autenticación de usuarios y almacenamiento en la nube, que pueden ser útiles en el desarrollo de aplicaciones móviles.**  **Realm: Realm es una base de datos móvil de alto rendimiento que utiliza un modelo de objeto para acceder a los datos. Realm es una excelente opción para aplicaciones móviles debido a su rápida ejecución y bajo consumo de memoria.**  **GraphQL: GraphQL es un lenguaje de consulta para APIs que permite a los clientes solicitar solo los datos que necesitan, lo que lo convierte en una excelente opción para aplicaciones móviles que tienen limitaciones de ancho de banda o recursos. GraphQL también es altamente personalizable y se puede utilizar en combinación con una variedad de tecnologías de base de datos.**  **En resumen, las tecnologías que incorporaría en mi desarrollo dependerían de las necesidades específicas de mi aplicación y de los requisitos de gestión de información que necesite cumplir. Es importante considerar aspectos como la eficiencia, la escalabilidad, la seguridad y la simplicidad de uso para elegir la tecnología adecuada.**   1. En los requerimientos para el desarrollo de un app nos exigen utilizar retrofit o volley para la gestión con una base de datos en MySQL que reside en un servidor. ¿De qué manera implementaría la integración con estos recursos?   **Para integrar Retrofit o Volley en mi aplicación móvil y realizar la gestión con una base de datos MySQL que reside en un servidor, seguiría los siguientes pasos:**  **Configuración del servidor: Debo asegurarme de que el servidor donde está alojada la base de datos MySQL permita la conexión desde la aplicación móvil. Debo crear una base de datos y una tabla que contenga la información que necesite gestionar.**  **Integración de Retrofit o Volley: Debo agregar las dependencias necesarias para utilizar Retrofit o Volley en mi aplicación móvil. Retrofit es una biblioteca que se utiliza para realizar peticiones HTTP y obtener datos de forma estructurada en formato JSON. Volley, por otro lado, es una biblioteca que permite realizar peticiones HTTP y obtener datos en diferentes formatos.**  **Definición de la interfaz API: Debo definir una interfaz API que especifique las operaciones que quiero realizar en la base de datos. En la interfaz, especificaré los endpoints del servidor que utilizaré para realizar las operaciones de CRUD (Crear, Leer, Actualizar y Eliminar). También puedo definir los parámetros que se pasarán en la petición y los formatos de los datos que se recibirán en respuesta.**  **Implementación de la lógica de negocio: Debo implementar la lógica de negocio en mi aplicación móvil para conectarme con el servidor y realizar las operaciones CRUD. La lógica de negocio es el código que se encarga de manejar los datos y las operaciones que se realizan en la base de datos MySQL.**  **Pruebas y depuración: Finalmente, debo probar mi aplicación móvil y depurar cualquier error que surja. Es importante asegurarse de que la integración con Retrofit o Volley y la base de datos MySQL sea correcta y que las operaciones CRUD se realicen correctamente.**  **En resumen, para integrar Retrofit o Volley en mi aplicación móvil y realizar la gestión con una base de datos MySQL que reside en un servidor, debo configurar el servidor, integrar las bibliotecas Retrofit o Volley, definir la interfaz API, implementar la lógica de negocio, y finalmente realizar pruebas y depuración para asegurarme de que todo funciona correctamente.** | | | | | | |
| **Presaberes Requeridos** | | | | | | |
| Utiliza el SQL y sus funciones como:   * CREATE DATABASE * CREATE TABLE * SELECT * INSERT * UPDATE * DELETE   Realiza consulta compuestas utilizando JOIN | | | | | | |
| **Marco conceptual o referencial \*** | | | | | | |
| **Enlace a Datos**  Esta guía muestra la forma de enlazar una colección de objetos a un objeto ***ListView***. Se muestra la forma de definir plantillas de datos para que el objeto *ListView*pueda tener conocimiento de cómo se deben presentar los elementos. **Trabajando con colecciones y el Control ListView** *ListView* es un control para presentar listas de datos, especialmente las listas largas que requieren desplazamiento.  *ListView* es rellenado con datos utilizando su propiedad ***ItemsSource*** que puede aceptar cualquier colección que implemente ***IEnumerable***. La manera más sencilla para rellenar un *ListView* implica el uso de un arreglo de cadenas como se muestra en el siguiente ejemplo.              Data = new List<string>              {                  "mono",                  "monodroid",                  "monotouch",                  "monorail",                  "monodevelop",                  "monotone",                  "monopoly",                  "monomodal",                  "mononucleosis"              };                listView.ItemsSource = Data;  El código anterior rellenará el *ListView* con una lista de cadenas. De forma predeterminada, *ListView* invoca al método *ToString* y muestra el resultado en un ***TextCell*** para cada fila.  Dado que ItemsSource se ha rellenado con una colección List<string>, el contenido no se actualizará conforme la lista cambie. Si deseamos que el *ListView* se actualice automáticamente conforme se agreguen, eliminen o modifiquen los elementos de la lista, debemos usar un ***ObservableCollection***. *ObservableCollection* está definida en *System.Collections.ObjectModel* y es exactamente igual que *List*, excepto en que puede notificar a ListView los cambios.              Data = new ObservableCollection<string>              {                  "mono",                  "monodroid",                  "monotouch",                  "monorail",                  "monodevelop",                  "monotone",                  "monopoly",                  "monomodal",                  "mononucleosis"              };                listView.ItemsSource = Data; **Enlace de datos y el control ListView** Enlace de datos es la característica que enlaza las propiedades de un objeto de interfaz de usuario a las propiedades de un objeto CLR.  Enlace de datos funciona manteniendo los objetos sincronizados a medida que cambian sus valores enlazados. En lugar de tener que escribir controladores de eventos para cada vez que cambia el valor de un control, establecemos el enlace y habilitamos el enlace en el objeto CLR.  Propiedades de celdas (y elementos hijos de las celdas) se pueden enlazar a propiedades de objetos en el *ItemsSource*. Por ejemplo, podríamos utilizar un *ListView* para presentar una lista de empleados.  La clase ***Employee***:          public class Employee          {              public string DisplayName { get; set; }          }  ***ObservableCollection<Employee>*** se crea y se establece como el *ItemsSource* del *ListView*.              ObservableCollection<Employee> employees =                  new ObservableCollection<Employee>();                employees.Add(new Employee { DisplayName = "Rob Finnerty" });              employees.Add(new Employee { DisplayName = "Bill Wrestler" });              employees.Add(new Employee { DisplayName = "Dr. Geri-Beth Hooper" });              employees.Add(new Employee { DisplayName = "Dr. Keith Joyce-Purdy" });              employees.Add(new Employee { DisplayName = "Sheri Spruce" });              employees.Add(new Employee { DisplayName = "Burt Indybrick" });                //listView definido en XAML              listView.ItemsSource = employees;  El siguiente fragmento de código muestra un *ListView* enlazado a una lista de empleados:      <ListView x:Name="listView">          <ListView.ItemTemplate>              <DataTemplate>                  <TextCell Text="{Binding DisplayName}" />              </DataTemplate>          </ListView.ItemTemplate>      </ListView>  El ejemplo anterior define un *ListView*. El origen de datos del *ListView* se establece a través de la propiedad *ItemsSource* con código C# (aunque podría haber sido establecido en XAML). El diseño de cada fila del *ItemsSource* es definido dentro del elemento ***ListView.ItemTemplate***.  **Enlazando SelectedItem**  Frecuentemente deseamos enlazar el elemento seleccionado de un *ListView* en lugar de usar un controlador de eventos para responder a los cambios. Para hacer esto en XAML, enlazamos la propiedad *SelectedItem*.          <ListView x:Name="listView"                    SelectedItem="{Binding SelectedEmployee}"> | | | | | | |
| **Actividad de Trabajo Autónomo** | | | | | | |
| Instale SQLite y revise su funcionalidad creando:   * Base de datos * Cree dos tablas * Cree una llave foránea que las relaciones | | | | | | |
| **Actividad de Comprobación del Trabajo Autónomo** | | | | | | |
| Cuadro comparativo de elementos de la actividad planteada | | | | | | |
| **Materiales, equipos e insumos a utilizar** | | | | | | |
| |  |  | | --- | --- | | **Materiales, equipos e insumos proporcionados por la Universidad** | | | Ítem | **Cantidad** | | *Computador por estudioso* | *1* | | *Conexión a Internet* | *1* | | *Android Studio* | *1* | | *Java* | *1* | | *Visual Studio Code* | *1* | | *Visual Studio 2022* | *1* |  |  |  | | --- | --- | | **Materiales del estudiante** | | | Ítem | **Cantidad** | | *N/A* |  | | *N/A* |  | | *N/A* |  |   El docente debe relacionar la cantidad de equipos y/o elementos necesarios para el desarrollo de la práctica, así como los que debe traer el estudioso. Es importante que previo al diligenciamiento de este apartado el docente verifique con la coordinación de laboratorios los recursos disponibles, esto con el fin de que la práctica se desarrolle de la manera más adecuada. | | | | | | |
| **Precauciones, nivel de riesgo y recomendaciones a considerar** | | | | | | |
| |  |  |  |  | | --- | --- | --- | --- | | **CLASIFICACIÓN DEL RIESGO** | **Muy alto** | | **Medio** | | **Alto** | | **Bajo** | | **FACTORES DE RIESGO** | | **CÓMO MINIMIZAR LOS FACTORES DE RIESGO** | | | * **Perdida de la información** | | * **Realizar copias de la información en drive externos** | | | **RECOMENDACIONES, CONSIDERACIONES PARA EL USO DE MATERIAL Y EPP** | | | | | En todo caso se deben seguir las siguientes recomendaciones para minimizar los riesgos propios de la actividad:  1. Identificar y conocer el protocolo de seguridad de laboratorios de informática.  2. No navegar en internet sin autorización del docente.  3. No ejecutar programas sin autorización del docente.  4. No instalar en los equipos Software de ninguna índole.  5. No trasladar equipos de cómputo de su módulo sin autorización del personal del área.  6. Cuidar sus objetos personales.  7. Cada alumno tiene como responsabilidad recibir las actividades de cada clase y apropiarse del material necesario para el desarrollo de las mismas.  8. Está prohibido el ingreso o consumo de alimentos, bebidas, chicle... dentro de la sala.  9. Está prohibido el uso e ingreso de dispositivos como celulares, parlantes y memorias USB sin autorización.  10. No conectar ni desconectar dispositivos como teclados, mouse o conexiones, en caso de anomalía avisar al profesor para realizar cambios o conexiones.  11. El trabajo debe hacerse en silencio, evitando las reuniones o interrumpiendo las actividades de otros estudiantes.  12. Cuide el buen funcionamiento del equipo que le ha sido asignado, evite cambiar configuraciones o intervenir los programas y propiedades del sistema operativo, el auxiliar de laboratorio es el único autorizado.  13. Todo dispositivo (teclado o mouse) que se pierda o se dañe con intención deberá ser repuesto.  14. No portar maletines o morrales, estos deben quedar depositados en los lockers destinados para ello. En caso de duda pida el respectivo candado con los auxiliares de cada laboratorio.  15. No rayar mesas, sillas, paredes y equipos, cuidar el aseo y orden de su puesto de trabajo.  16. Se prohíbe el ingreso o exploración de páginas no autorizadas y pornográficas, es causal de sanción y expulsión (vetado) de la sala de informática por varias sesiones. | | | | | **CONSIDERACIONES ÉTICAS** | | | | | * En caso de ser necesarias. | | | |   Respetar los derechos de autor | | | | | | |
| **Procedimiento y Metodología de la práctica** | | | | | | |
|  | | | | | | |
| **Criterios de Entrega – Informe de Laboratorio** | | | | | | |
| Exposición en el idioma inglés de la actividad realizada. – Soporte de video del desarrollo de la actividad | | | | | | |
| **Criterios de Evaluación – Práctica de Laboratorio** | | | | | | |
| |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **RUBRICA DE EVALUACIÓN DESARROLLO DE SOFTWARE SEGURO** | | | | | | | | | | | | | **INDIVIDUAL** | | | | | **CRITERIOS DE EVALUACIÓN** | | | | | **NOTA** | | | **Habilidad** | | **Estudiante** | | | **0 – 1,5** | **1,6 - 2,9** | **3,0 - 3,9** | **4,0 - 4,5** | **4,6 - 5,0** | |  | | |  |  |  |  |  |  | | |  | | |  |  |  |  |  |  | | | **GRUPAL** | | | | **%** | **0 – 1,5** | **1,6 - 2,9** | **3,0 - 3,9** | **4,0 - 4,5** | **4,6 - 5,0** | **NOTA** | | | **competencia** | **Procedimental**  **(aprender a hacer)** | | Organiza los resultados obtenidos a través del uso de gráficas, dibujos, tablas, mapas conceptuales.  Documenta y organiza el código fuente y demás recursos utilizados para la solución del problema planteado. | **35%** | No son precisos o no ayudan a la comprensión del tema. | Son organizados y algunas veces ayudan a la comprensión del tema. | Son organizados y ayudan al entendimiento del tema. | Son precisos y ayudan a la comprensión del tema. | Son organizados, precisos y ayudan al entendimiento del tema |  |  | | Hace uso adecuado de los conceptos al momento de aplicarlos en la práctica experimental propuesta. | Se limita a la recopilación de la información solicitada. | Hace referencia a la información en la práctica. | Hace relaciones básicas de la información con la práctica. | Analiza la información relacionada con la práctica. | Infiere la información obtenida y la relaciona con su realidad. |  | | **Cognitiva**  **(aprender a conocer)** | | Utiliza correctamente los tipos de datos, declaración de variables, funciones, estructuras de control condicional y repetitivas para la solución de un problema. | **35%** | Se le dificulta la declaración de variables, el uso de funciones y de estructuras de control. | Declara variables. Se le dificulta el uso de funciones y de estructuras de control. | Declara variables, utiliza funciones. Se le dificulta el uso de estructuras de control. | Declara variables, utiliza funciones y estructuras de control. | Utiliza correctamente la declaración de variables, el uso de funciones y de estructuras de control. |  |  | | Comprende el concepto y uso de las estructuras de datos en ingeniería. | No comprende el concepto, uso y aplicaciones de las estructuras de datos en ingeniería. | Comprende el concepto de estructuras de datos. No identifica su uso y aplicaciones. | Comprende el concepto de estructuras de datos y algunas aplicaciones de las mismas. | Comprende el concepto, uso y aplicaciones de las estructuras de datos en ingeniería. | Comprende el concepto, uso y aplicaciones de las estructuras de datos en ingeniería y en su profesión. |  | | **Socio afectiva**  **(aprender a ser)** | | Desarrolla habilidades de trabajo en equipo, priorizando la toma de decisiones y la escucha de diferentes propuestas. | **10%** | No hace parte del trabajo propuesto por el equipo. | Parcialmente hace parte del trabajo propuesto por el equipo. | Hace parte del trabajo propuesto por el equipo de acuerdo a parámetros básicos. | Participa en el trabajo propuesto por el equipo de manera responsable y puntual. | Participa activamente el trabajo propuesto por el equipo de manera responsable y puntual. |  |  | | Cuida, respeta y exige respeto frente a la interacción con sus pares y docentes. | Frecuentemente reprocha el trabajo de sus pares y docente, y justifica sus carencias en el trabajo en grupo. | A veces muestra una actitud favorable frente a la clase y se limita a responder por las condiciones básicas del trabajo. | Muestra una actitud favorable frente a la clase e interactúa ocasionalmente con sus pares y docente. | Muestra una buena actitud frente a la clase e interactúa con sus pares y docente. | Muestra una sobresaliente actitud frente a la clase e interactúa frecuentemente con sus pares y docente. |  | | **Comunicativa**  **(aprender a convivir)** | | Practica el uso de lenguaje escrito como medio de identificación y diferenciación en la elaboración de informes. | **10%** | No construye el informe de laboratorio. | Construye el informe de laboratorio de manera incompleta. | Construye el informe de laboratorio de acuerdo a los requerimientos mínimos. | Construye de buena manera el informe de laboratorio. | Construye de manera sobresaliente el informe de laboratorio. |  |  | | Utiliza lenguaje técnico para referirse a los diferentes conceptos que relaciona en la práctica experimental. | No hace uso de un lenguaje técnico apropiado para la práctica de laboratorio. | Ocasionalmente hace uso de un lenguaje técnico apropiado para la práctica de laboratorio. | Hace uso de un lenguaje técnico apropiado para la práctica de laboratorio. | Hace un buen uso del lenguaje técnico apropiado para la práctica de laboratorio. | Hace un uso sobresaliente del lenguaje técnico apropiado para la práctica de laboratorio. |  | | **Investigativa** | | Realiza la búsqueda bibliográfica en fuentes confiables que permitan dar respuesta a las situaciones problema evidenciados en la consulta previa y en el informe de laboratorio. | **10%** | Las fuentes de información son pocas o ausentes. Si las usa son poco confiables y no contribuyen a la construcción del eje central. | Las fuentes de información son restringidas. No son actualizadas y contienen información poco relevante. | Las fuentes de información son relevantes e informativas. Presenta los parámetros aceptables por el docente. | Las fuentes de información son variadas e informativas. Adicionalmente son fiables y contribuyen al tema. | Las fuentes de información son variadas y pertinentes. Además están actualizadas y contienen información relevante al tema. |  |  | |  | Consulta el estado del arte sobre computación paralela y elabora un informe sobre API para computación paralela. | No realiza la consulta ni el informe. | Realiza la consulta pero no presenta el informe. | Presenta el informe de manera incompleta. | Presenta el informe. | Cumple con los requerimientos del informe. |  | | | | | | | |
| **Palabras Clave** | | | | | | |
|  | | | | | | |
| **Bibliografía Recomendada** | | | | | | |
| Muñoz Rodríguez D. (;2002).Sistemas Inalámbricos de Comunicación Personal. Alfaomega Grupo Editor. (Catálogo biblioteca UMB)  Friesen J, Dans Álvarez P.(2010). Java. Para Desarrollo Android. Ediciones Anaya Multimedia.(Catálogo biblioteca UMB)  Napier R, Kumar M, Salcedo Sotoca FJ.(2013). IOS 6. :Desarrollo de Aplicaciones. Ediciones Anaya Multimedia.(Catálogo biblioteca UMB) | | | | | | |
| |  |  |  | | --- | --- | --- | | **Tema** | **Subtema** | **Referente bibliográfico** | | Control de código fuente | Control de versiones | Liles, S. (2013). Asynchronous Android : Harness the Power of Multi-core Mobile Processors to Build Responsive Android Applications. Birmingham, UK: Packt Publishing. Retrieved from <http://proxy.umb.edu.co:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=679959&lang=es&site=eds-live&scope=site> | |  | | Código fuente | Bin Aftab, M. U., & Karim, W. (2014). Learning Android Intents. Birmingham, UK: Packt Publishing. Retrieved from http://proxy.umb.edu.co:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=690403&lang=es&site=eds-live&scope=site | |  | |  | Team Foundation Server | Aydin, Murat. 2012. Android 4 : New Features for Application Development. Community Experience Distilled. Birmingham: Packt Publishing. http://proxy.umb.edu.co:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=526160&lang=es&site=eds-live&scope=site. | | | | | | | |
| **Fecha de Actualización** | | | | **Descripción** | | **Participantes** |
| Diciembre de 2022 | | | |  | | Robert Osorio Torres |