SYLLABUS

Course	Mobile App Development (Tipo i)
Code	ISIS-3510
Department	Systems and Computing Engineering
Semester	2023-20
Lectures	**Section 1:** Monday , Wednesday and Friday: 8:00 am – 09:20 am* **Section 2:** Monday , Wednesday and Friday: 9:30 am – 10:50 am* **Section 3:** Monday , Wednesday and Friday: 11:00 am – 12:20 pm*
Office hours	**Mario** *Monday 3:00 pm – 4:00 pm (ML 704), Thursday 2:00 pm - 3:00 pm (ML 704)* **Camilo** *Tuesday 11:00 am – 12:00 pm (ML 315), Thursday 2:00 pm – 3:00 pm (ML 315)* **Nicolás** *Monday 2:30 pm – 3:30 pm (ML 755), Friday 9:30 am – 10:30 am (ML 755)* **Laura** *Wednesday 10:00 am – 12:00 pm*** **Erik** *Monday 11:30 am – 12:30 pm**, Tuesday 3:00 pm – 4:00 pm*** **Andrés Felipe** *Tuesdays 8:00 am – 10:00 am*** **Jairo** *Wednesday 8:30 am – 10:30 am*** **Maria Paula** *Monday 5:00 pm – 6:00 pm**, Wednesday 5:00 pm – 6:00 pm*** **Luisa** *Tuesday 10:00 am – 12:00 pm*** *Attending lectures is mandatory. We follow the attendance policies established by the University ** DODWTA: Defined on-demand with the TA

Lecturers

Name	Email
Mario Linares Vásquez Camilo Escobar Velásquez Nicolás Cardozo	m.linaresv@uniandes.edu.co ca.escobar2434@uniandes.edu.co n.cardozo@uniandes.edu.co

TAs (Teaching Assistants)

Name	Email	Platform
Jairo Adolfo Céspedes Plata	ja.cespedes@uniandes.edu.co	Flutter
Luisa Fernanda Fuentes Ladino	l.fuentesl @uniandes.edu.co	Kotlin
Andrés Felipe Lugo Saavedra	a.lugos@uniandes.edu.co	Kotlin
Maria Paula González Escallón	m.gonzaleze@uniandes.edu.co	Flutter
Laura Andrea Rodríguez Rodríguez	la.rodriguez@uniandes.edu.co	Flutter
Erik Nielsen Rodríguez	ea.nielsen@uniandes.edu.co	Kotlin

1. Course Description

The main goal of this course is to promote/generate in the students, the skills required to conceive, design, build, and test software solutions based on mobile platforms, by following a development process that considers mobile app development-specific challenges and design constraints. Therefore, during the course, the students will explore concepts related not only to mobile devices, but also infrastructures, environments, and software lifecycles.

"Tipo i". Note that this is a "tipo i" course. Thus, ISIS3510 focuses also on (i) developing communicative competencies (receptive and productive) in English that are relevant for academic and professional performance; and (ii) acquiring academic and content language through exposure to authentic materials in diverse formats (print, audio-visual, etc.) in English.

This course is taught from the perspective of Software Engineering, Design, and Architecture, having as main drivers the design and implementation of context-aware systems considering design constraints and in particular two quality attributes: performance and eventual connectivity. As the main deliverable of the course, the students have to implement a real product (i.e., a software solution that provides value) following an iterative and incremental process; the iterations in the process are aligned with the course modules in such a way that the product from each iteration has to include the concepts presented/explored in the corresponding module. Note that the course focuses on design, software engineering, and architecture. **This is not a programming course.**

The mobile development process performed by the students includes the conception, design, programming, and testing tasks, and aims at building a software solution that includes apps deployed in mobile devices by using two natives from the following list: Kotlin, Flutter/Dart, Android-Java, & iOS[1]. To provide the students with the knowledge of how to use the mobile

[1] Usage of hybrid cross-platform tools is not allowed; React Native is not allowed

development APIs and platforms, we have prepared lists of resources they can check to learn mobile programming concepts on their own.

It is worth noting that at the end of the course, the students should release a functional app that includes all the concepts presented in the lectures and labs. In addition, this course will explore architectures and internals of the mobile operative systems, which is relevant content to understand several performance and security issues in mobile apps.

At the end of the course, a student should be able to:

•**O1:** Design mobile apps by considering design constraints and in particular two quality attributes: performance and eventual connectivity

•**O2:** Build native mobile apps that provide users with a multi-device experience, and use basic programming components, sensors, location services, and data storage.

•**O3:** Justify design and implementation decisions based on the specifics/particularities in (i) the mobile ecosystem, (ii) the development process, and (iii) the dominant mobile platforms.

•**O4:** Use tools that support software development tasks such as design, development, and resource profiling/monitoring.

2. Methodology

This course syncretizes different methodologies such as flipped learning/classroom, challenges-based learning, and incremental learning but with a special focus on project-based learning. A project is the main component that serves as the root and final goal of every activity in the course. This course content is organized into 4 teaching modules that present/explore specific topics in an incremental and complementary way. Each one of the modules includes activities oriented to "practice" the learned concepts and develop the project iteratively. Therefore, a learning module is also an iteration/sprint.

A sprint represents a clear "challenge" that working teams must achieve; each challenge imposes specific design constraints and service level agreements that are directly related to the corresponding learning module. A challenge is also an iteration in the "pure" agile philosophy, with a defined time window, and the expectation of a complete product to be delivered. Each sprint is divided into weekly micro-sprints that are designed to help the teams to accomplish a challenge. Because we follow an incremental model, the products of each micro-sprint are small steps for building the product expected to have for the corresponding sprint.

In summary, each sprint can be defined as a tuple S = <challenge, learning module, microsprints>.

Concerning the relationship between challenges and flip-flop learning, the design of this course carefully defines a set of activities that **emphasizes student autonomy and discipline**. There will be outside-class activities that must be finished by the students before each class; examples of these outside-class activities are on-the-field data collection tasks, and reading our online book (yes, we have an online book). Remember that this is also a challenge-oriented course in which solutions to the challenges (i.e., each sprint) are designed by the students. This is a shift from the traditional lectures the students are used to having. The classes will be devoted to lectures, teamwork, and feedback sessions, because the students must prepare the class a-priori and be ready to discuss the class content. We will have a few traditional lectures because some topics

are more suitable for a traditional lecture. Anyway, the classes are expected to be more interesting, dynamic, interactive, and focused on the specific issues the students experience with the outside-class activities.

Classes will be used to consolidate the concepts of each micro-script, work in teams and get feedback from the lecturers and TAs. In addition, the students can receive continuous support with their English language learning from the English tutoring center at the Coffee Time place that is located on the second floor of the Santo Domingo building. Therefore, all the deliverables generated by the students, and communications with the course team should be in English.

This course requires from the students, discipline and compromise with the learning activities.

3. Content and Modules

This is the capstone project of the "ISISYC" program; this is not a programming course; this is a software engineering and architecture course. Therefore, we will not teach you the language syntaxes or the specifics of each language. There are courses and books specifically designed to teach you how to program Android, Kotlin, or iOS apps, and we encourage you to go there if you want to be a specialist programmer.

As mentioned before, this course content is organized into 4 teaching modules that present/explore specific topics in an incremental and complementary way. Each one of the modules includes activities oriented to "practice" the learned concepts. These activities are expected to help the student to explore/learn in a guided and autonomous way the main concepts and topics regarding mobile app development

The content modules are described as follows:

***Module 1 – Powerful Context (Aug 8th – Sep 2nd):**
- Design thinking philosophy.
- The mobile ecosystem and types of mobile apps
- Mobile App life cycle and app conception
- UX/UI in mobile apps; GUI metaphors and components
- Value proposition

* **Module 2 - Design and … more design (Sep 2nd – Sep 30th):**
- Architecture concepts recap.
- The architecture of the Mobile Operating Systems
- Programming model and components: Android, Kotlin, iOS
- Architectures and patterns for mobile applications

* **Module 3 – The quality attributes matter (Oct 7th – Nov 4th):**
- Policies (OSes and Frameworks) that impact mobile app performance
- Local and external storage
- Best practices for eventual connectivity management
- Asynchronous services, threads, backend services, and background services

* **Module 4 – Enhance your app (Nov 4th – Dec 2nd):**
- Best practices for performance
- Tools for profiling, monitoring, and linting

- Vulnerabilities in the mobile operating systems
- Malware and vulnerabilities in mobile apps

4. Grading

The grading scheme is based on the following principles

- **This course aims at developing and promoting team-related and individual competences**
- **Students always will have a clear view of their learning process and the correlation with grading**
- **Flexibility for promoting autonomy, self-paced, and incremental-learning**

Following those principles, we designed a grading system based on points that are earned with different activities such as midterms, in-class activities, homework, project deliverables, etc. The activities will assess competencies in three different levels based on the Bloom taxonomy for educational objectives and inspired by the SFIA framework: (i) explain ideas and concepts, (ii) use information in new situations, (iii) justify a stand or decision, and (iv) produce new or original work.

Therefore, during the course, the students will have the following distribution of maximum available points grouped by activities:

TP = "team points", IP = "individual points"

Activities	Max points
Midterm 1	61 IP
Midterm 2	120 IP
Final exam	135 IP
Sprint 1	180 TP, 245 IP
Sprint 2	180 TP, 170 IP
Sprint 3	180 TP, 165 IP
Sprint 4	180 TP, 205 IP
App report - Sprint 1	15 IP
App report - Sprint 2	45 IP
App report - Sprint 3	75* IP
App report - Sprint 4	45* IP
Total	720 TP, 1281 IP

*** These points are optional**

Each activity that provides points to the students will have a detailed assessment matrix explaining the evaluated competencies and how the points are assigned based on the student's deliverables, answers, etc.

Note that App report – Sprint 3, and App report – Sprint 4 are optional activities. A caveat: for those teams that decide to present the App report – Sprint 4, outstanding reports will allow students to skip the final exam.

The complete list of competencies and points that a student can earn on each one are listed here: https://tinyurl.com/5xw74bpj

The points will be translated to the final grade by using the following table:

FINAL GRADE	Required to get 3.0	Required to get 4.0	Required to get 5.0	Max. available
Individual points	700	850	1000	**1281**
Team points	600	650	700	**720**
TOTAL	**1300**	**1500**	**1700**	**2001**

4.1. Activities - Individual Points

- Two mid-terms + One exam
- Project: it includes 4 development iterations aligned with the 4 teaching modules; therefore, each iteration focuses on different quality attributes. A project is developed by teams composed of 6 students. At the end of each iteration (except for the initial iteration that is devoted to conceiving the app idea) a working product must be delivered in two native platforms. The students will follow a rotating-leader scheme, i.e., each iteration will have a different student leading the team. Remember that as part of the deliverable, students will release a version of their apps for testing and should present their progress. The language in the app must be English, and the presentations should be delivered also in English.
- "Viva voce[2]": in addition to sprint deliverables, that provide individual points to each student but are earned as a team (i.e., the deliverable grade is the team members' grade), students ---each sprint --- can get individual points in "live sessions" with the lecturers. The idea of the "Viva voce" sessions is to evaluate competencies individually; each student decides what competencies want to demonstrate, with the purpose of earning points. The list of individual competencies for each sprint is detailed in the competencies spreadsheet: https://tinyurl.com/5xw74bpj
- Micro-sprint activities: refer to quizzes, lecture controls, and class preparation/participation. In the weekly micro-sprint plan, we will let you know what activities are graded.
- Bonuses: additional tasks will be suggested during the semester. Those tasks are not mandatory but will have extra points over the final grade.
- English Language component: We will evaluate your writing and speaking skills as a component of (i) the sprint deliverables (apps, slides, and presentation), and (ii) an app audit report you must write as part of the final exam. In both cases, sprint deliverables and audit report, you will count on the support of the English tutoring center.
    - Assessment of reading skills is already implicit in all the activities because all the material is in English (guides, lectures, tutorials, etc.)

[2] https://www.merriam-webster.com/dictionary/viva%20voce

The delivered product, the process, and peer assessments are components of each iteration grade. The quality of the product and process is evaluated based on (i) the artifacts (i.e., repository, documentation, etc.), (ii) the expected features, requirements and constraints of each iteration, (iii) individual contributions, and (iv) individual presentation of the product. The product (at the end of each iteration) is expected to work properly on real devices.

4.2. Activities - Team Points

Co-evaluation. This is a mechanism that allows students to provide us with anonymous feedback about the performance of each member of a team concerning a given sprint deliverable. At the end of each sprint, you have to evaluate your partners **HONESTLY** based on a pre-defined set of team competencies that are listed in our competencies sheet: https://tinyurl.com/5xw74bpj. NOTE that the points provided in the coeval are a proxy for the team points, i.e., that we (lecturer and TAs) will verify the validity and trustworthiness of the points provided by each student to her teammates. **If we find inconsistencies between the points reported by one student and the real contribution of one of her teammates, then, the team points of the student will be penalized. If in a team we find inconsistencies in the points reported by more than two students, then the whole team will be penalized.** The co-evaluation is a mandatory task.

For the co-evaluation, you have to fill out an online form and follow these rules:

- If you do not fill out the co-evaluation by the established deadlines, your points are going to be equal to the points assigned to you (by your partners) but multiplied by a penalization factor *P* using the following formula: points = team points * (1-*P*). This factor increases on each sprint: for Sprint 1, *P* = 0.05; for Sprint 2, *P*=0.1; for Sprint 3, *P*=0.15; and for Sprint 4, *P*=0.2

- If only one member of a team submits the co-evaluation, that person will score the complete points, and the other members will have the aforementioned penalization.

- If nobody in a team submits a co-evaluation in the established times, then the team points for the corresponding sprint will be zero.

- **No deadline extensions will be granted. Students have to pay attention to the TAs messages at teams informing the co-evaluation deadlines.** The co-evaluation deadlines will not be modified. The times for filling out the co-evaluations are as follows:

Sprint	Co-evaluation time
Sprint 1	August 8th (05:00 am) – September 2nd (5:00 am)
Sprint 2	September 2nd (05:00 am) – September 30th (5:00 am)
Sprint 3	October 7th (05:00 am) – November 4th (5:00 am)
Sprint 4	November 4th (05:00 am) – December 2nd (5:00 am)

Ethics component. At the end of each sprint, each team should deliver (in the team WIKI) a video, no longer than 6 mins, in which each member makes a reflection concerning (i) the ethical implications of her work as part of the team, (ii) the societal and environmental impact of the proposed product, (iii) ethical implications of the product. This is a mandatory task, and the participation of all the members of a team must be visible in the video.

Multiplier. The total team points of a sprint co-evaluation (for a student x in team y) will be calculated as follows: team-given points *(x) * multiplier(y)*. This multiplier is a number given by the ISIS3510 team based on the team performance (e.g., contribution, git-flow). For example, Mario got 134 team points (from his teammates) in sprint 1, and his team did a pretty good job and obtained a multiplier equal to 1. Then, his total team points for sprint 1 will be 134 * 1 = 134. The ISIS3510 team computes the multiplier as follows. **Each team member should report their contributions in the sprint wiki, as part of the sprint deliverable. For example, Ana adds to the WIKI the following:**

- Anamaria Mojica's contributions.
  - Implement View x (file.x, file.y) in Kotlin app
  - Create Context canvas and ……
  - …

Then, taking into account the information provided by each member of a team and the information provided by each team repository (e.g., milestones, issues), the ISIS3510 team will grade each team member using the following rubric, and the average of all the members of the team will be the final multiplier.

Criteria	Milestones	Opens issues	Closes Issues	Pull request – approval/closed	Pull request	Does commits	Equitable contribution
None	0	0	0	0	0	0	0
Partially Accomplished	0.071	0.071	0.071	0.071	0.071	0.071	0.071
Accomplished	0.14	0.14	0.14	0.14	0.14	0.14	0.14

For example, team 01 is composed of Jane Doe and Joe Doe. Jane Doe scores "**Partially Accomplished**" in all the components (in the contributions rubric), which gives her a total of 0.5 (0.071+0.071+0.071+0.071+0.071+0.071+0.071). In addition, Joe Doe scores "None" in the first three components and "**Partially Accomplished**" in the other four components; that gives him a score of 0.28 (0+0+0+0.071+0.071+0.071+0.071). On average, the group had a multiplier of 0.39 ([0.5 + 0.28]/2), which means that the team points will be multiplied by 0.39.

TIP: make sure that you do not have connectivity issues, and after filling out the form you see the Microsoft Forms confirmation window saying "Your response was submitted". We also recommend you get a copy of the answers via the "Print or get PDF of answers" (in the same window), so you have evidence in case a complaint needs to be filed to the ISIS3510 team.

4.3. Other grading guidelines

- Final grades will be in the range [1.50, 5.00], with a precision of two decimal figures. In the cases of grades with more than two decimal figures, classic rounding will be applied to the decimal figures in order to obtain two decimal figures.

- This course is approved with a final grade of 3.00 / 5.00. There is no automatic approximation, e.g., there is no approximation from 2.99 to 3.00.

- Each student could recover points for the Sprint 1-3 viva voce. To do this, each student will have up to the last day of the next sprint, for example, if the student Mario Linares

wants to recover points for his Sprint 1 viva voce, he could do it up to October 1st. Each student interested in recovering points has to make an appointment with the lecturer, based on his availability, so make sure that you book the appointment in advance.

- **Last day for earning points: December 11th, 2023**

4.4. Sprints and App reports deadline, and Midterms schedule

Sprints	• Sprint 1 deliverable: September 9th, 2023 (5th week) • Sprint 2 deliverable: September 30th, 2023 (8th week) • Sprint 3 deliverable: November 4th, 2023 (12th week) • Sprint 4 deliverable: December 2nd, 2023 (last week)
App report	• App report 1: September 2nd, 2023 (4th week) • App report 2: September 23rd, 2023 (7th week) • App report 3: October 28th, 2023 (11th week) • App report 4: November 25th, 2023 in order to be considered to avoid final exam, otherwise, December 2nd, 2023
Midterms/Exam	• Midterm 1: September 8th, 2023 (5th week) • Midterm 2: November 10th, 2023 (13th week) • Final exam: December 1st 2023 (last week)

5. Resources

The course activities will be supported by the following infrastructure:

- An online virtual classroom at "Bloque Neón" plus (https://bloqueneon.uniandes.edu.co/) that contains the schedule, activities, micro sprint plans, sprints, etc.
- The main communication channel is Microsoft Teams. We created a channel for each type of activity.
- An online **Gitbook** with content that students must read and work with as part of the outside-class activities
- **Coffee-time:** English online tutoring environment, where students of "Tipo i" courses can receive continuous support with their English language learning. (https://sites.google.com/view/uniandescoffeetime/home)
- The teams will use Git as the version control system (VCS) and GitLab/GitHub for repository management. Each team is responsible for creating its private repository and giving "Reporter" privileges to the professor and lab instructors. The teams must use the GitLab/GitHub features for project management, bug/issue tracking, continuous integration, wiki, and pages.